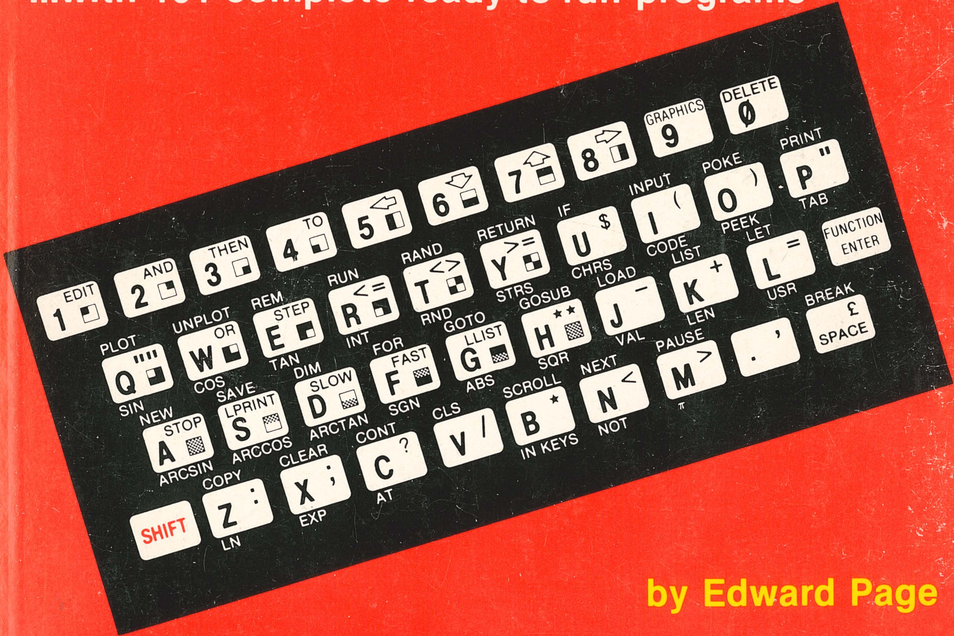
$7.95

101 TIMEX 1000 / SINCLAIR ZX-81 PROGRAMMING TIPS & TRICKS

The software ideabook, overflowing with hints, secrets, shortcuts and techniques for using the TIMEX/Sinclair 1000 and Sinclair ZX-81 computers ...with 101 complete ready-to-run programs



by Edward Page

101 TIMEX 1000 / SINCLAIR ZX-81 PROGRAMMING TIPS & TRICKS

101 TIMEX 1000 / SINCLAIR ZX-81 PROGRAMMING TIPS & TRICKS

by Edward Page

ARCsoft Publishers
WOODSBORO, MARYLAND

FIRST EDITION
FIRST PRINTING

Printed in the United States of America

Library of Congress (LC) number: 82-18413

ISBN 0-86668-020-9

Preface

The Sinclair ZX-81 and Timex/Sinclair 1000 are the most popular personal computers in the world. As of this writing more of those models have been sold than any other computer in history.

The very light weight and highly portable design of Sinclair's computer, coupled with the powerful version of the BASIC programming language it uses, have placed it at the very forefront of the new wave of computers for home, school and office.

Anything but a toy, the hardware configuration of the T/S 1000 and ZX-81, and its versatile system software, make it a highly useful tool in the business environment and the classroom as well as for practical jobs around the home.

In fact, the system software is so flexible that the need for this book became apparent. There are so many computer tasks which can be accomplished with the ZX-81 and TS/1000 that an introduction to the many techniques is needed.

Software programs are what make a computer do work for you. This book is written for those newcomers and beginners, as well as advanced novices and student programmers, who would like to tap the seemingly-unlimited resources in the T/S 1000 and ZX-81 computer package. It is hoped this book will guide and instruct and provide insights into the many ways the BASIC language in Sinclair's machine can be put to use.

This book is a companion volume to *37 TIMEX 1000/Sinclair ZX-81 Programs for Home, School & Office.*

—Edward Page

Table of Contents

Text on Text

Gee Whiz

Number Chrunching

Money Matters

Graphics

Appendix

Introduction

There is a great need for practical, useful software for the new generation of popular personal computers. The Sinclair ZX-81 and TIMEX/Sinclair 1000 computers, for example,are among the world's most popular. In fact, based on numbers sold they must be *the* most popular.

The T/S 1000 and the ZX-81 are the same computer, except the T/S 1000 has more built-in memory than the ZX-81. All 101 progams in this book have been written and tested on both computers and will run in the more-limited ZX-81 memory size. In other words, all of the programs in this book will run on either machine.

The T/S 1000 and the ZX-81 are powerful and versatile and flexible—but what can they do? Once you've purchased the hardware, you need down-to-earth ideas and workable programs to make the computer work for you.

The aim of this book is to provide more than 101 new and different ideas about how to use the computer and more than 101 complete, ready-to-type-and-run programs you can put to use immediately.

The 101 programs in this handbook can stand alone or be used as parts of larger sets of instructions you might write. These are designed to be typed in, just as you find them in this book, with no other programming needed.

These programs are useful in themselves. They also make good starting points for further development as you learn more about how to program your own computer and start to write more and more of your own programs. You can use these fun and practical programs and then modify them and expand them to suit your needs as they grow.

This book can be used by newcomers and beginners as well as by novices and students and more-advanced folks who are in need of new ideas. Amidst our 101 tips are many, many secrets, shortcuts, tricks, hints, techniques and make-it-easier instructions, each designed to make you a more versatile programmer and to make your programming effort lighter. Use this book to stimulate your thinking about how to approach various software problems. Use it to get good ideas for new and different approaches to all your programming goals.

We make the assumption that you know how to set up your T/S 1000 or ZX-81 (or MicroAce) for use. You know how to connect the cable between the TV switchbox and the computer. You know how to plug the power supply into a wall socket and attach the power cord to the computer. You know how to type on the flat keyboard, locate the shift key, and obtain the various "levels" of functions on the multiple-function keys. If you don't have these areas of skill down pat, yet, check your owner's manual. The Sinclair ZX-81 manual is entitled, *ZX-81 BASIC Programming* and is thorough. The similar publication which TIMEX ships with the T/S 1000 is the *TIMEX User Manual.* Please review these thoroughly as you start to use the programs in this book.

You *do not have to be a programmer* to use the software in this book. Just type in the programs as you find them here and run them. They will work!

A major part of your learning from this book will come as you go through the work of typing in these programs making sure you have each line typed correctly. If a syntax error, or other error, occurs, go back and make sure your typed line matches what we have printed in this book. Our program lines are believed to be exactly correct. All have been tested on a T/S 1000 and a ZX-81 and work properly.

If, after typing in a program line as we have it, you get an error message from your computer, check the handy list of error messages in the Appendix at the rear of this book. You also will find a convenient list of BASIC words, as used in the T/S 1000 and ZX 81, in the Appendix.

If you are getting an error message, most likely you will find you have made a typographical error in typing the program into the computer. However, should you find a typo which we may have made in this book, please let the author know. Send a postcard or letter to the author in care of *ARCsoft Publishers*, P.O. Box 132, Woodsboro, MD 21798 USA. The author will appreciate being able to make any necessary corrections to future editions.

Home, school and office

The book has been organized into several sections plus an appendix. The first section includes tips and tricks which you may find applicable to writing and playing games with your computer.

Other sections will advance your knowledge of how the computer handles text, works with numbers, creates graphics designs on the TV screen, etc. Naturally, the sections of the book, as divided, are not rigid and exclusive. You probably will find something in the graphics section to add to a "money matters" program. Or a "text" tip could easily be applicable to a game. You might find lots of additional "gee whiz" programs in other sections of the book.

Try them all. They're great fun to run. And they are especially designed to be short so you won't have to spend hours typing in one program.

REMarks

As you read through the 101 programs in this book, you will notice few REM, or remarks, statements. The author's training in writing BASIC-language computer programs included an emphasis on brevity and saving memory space. A sharp editing pencil was in order—and still is!

With only one to two kilobytes (1k to 2k) of memory available in the ZX-81 or T/S 1000, you need to create lean programs. There is no room for fat—and this is good! Programmers who have learned on such systems will make better professionals because they know how to write efficient "code."

REMarks and explanations in software are out. Honing, fine-tuning and waste trimming are in. Use of coding form program worksheets, such as the *TIMEX 1000/Sinclair ZX-81 BASIC Coding Form* published by ARCsoft Publishers, is encouraged. The objective always is to make the most efficient use of available memory.

Always remember: even though they may be headed toward the same goal, no two programmers will write exactly the same list of BASIC instructions, or program lines, from scratch. As you load these 101 programs into your T/S 1000 or ZX-81 computer, one at a time, you will make modifications to suit your personal needs and interests. Exact wording of PRINT statements, for instance, can be changed. Or, two or more programs can be combined into one grand scheme. Your application may vary.

If you want to load more than one of these programs into your T/S 1000 or ZX-81 at the same time, be sure to use different sets of line numbers for different programs. Two programs can't occupy the same set of line numbers at the same time!

ENTER vs. RETURN

Computer programmers today generally mix the two words, RETURN and ENTER, together and use them to mean the same thing. In either case, when you see RETURN or ENTER in this book, we mean the ENTER key on the right-hand side of the ZX-81 or T/S 1000 keyboard.

These programs will run on *any* computer which can be programmed in BASIC. However, to run on machines other than the T/S 1000 or ZX-81, you will have to make slight modifications to program lines. Graphics commands, especially, will differ on non-Sinclair computers.

Refer to the owner's manual which came with your non-Sinclair computer and compare its version of BASIC with the Sinclair BASIC. A list of Sinclair BASIC words can be found in the Appendix at the end of this book.

Also, if you use a computer other than T/S 1000 or ZX-81, such things as line numbering, logical tests, multiplication symbols, print statements and other instructions may differ.

Report codes

Sinclair refers to its error messages as *report codes*. We refer to them as error messages because that's how most programmers today would refer to them. Report codes and error messages are the same thing in this book.

Many of the programs in this book will continue to run until you command them off via the BREAK key. You may stop any run, at any time, with the BREAK key at the lower right-hand side of the keyboard.

The author would like to have your suggestions for changes in future editions of this work, or for other books in this series for the TIMEX/Sinclair 1000 and Sinclair ZX-81 computers. He may be addressed in care of ARCsoft Publishers.

Good programming!

Fun & Games

1 Coin Toss

Here's a handy way to settle arguments. Toss a coin. Only this time, let the computer do the work!

Type in the program. Run it. The computer will report *heads* or *tails* after each toss.

For a new toss, press any key on your computer's keyboard.

Line 20 clears the screen. A random number is generated at line 30.

Lines 60 and 70 accomplish the restart when you press any key.

Program Listing

```
10 RAND
20 CLS
30 LET R=RND
40 IF R<.5 THEN PRINT "HEADS"
50 IF R>.5 THEN PRINT "TAILS"
60 IF INKEY$="" THEN GOTO 60
70 GOTO 20
```

2 Traditional Dice Roll

Here's a simple, brief way to roll and display results for two dice.

Lines 100-110 get a random number between 1 and 6 and store it in A. Lines 200-210 get another random number from 1 to 6 and store it in B.

Lines 300-310 print the contents of A and B along with a suitable message. Lines 400-420 cause the computer to await the press of any key on its keyboard. When a key is pressed, dice are rolled and new results presented on the video display.

Program Listing

```
 10 RAND
100 LET A=INT(10*RND)
110 IF A<1 OR A>6 THEN GOTO 100
200 LET B=INT(10*RND)
210 IF B<1 OR B>6 THEN GOTO 200
300 PRINT "FIRST DICE: ",A
310 PRINT "SECOND DICE: ",B
400 IF INKEY$="" THEN GOTO 400
410 CLS
420 GOTO 100
```

3 See Two Dice

Here's a quick way to add real dice to any fun program you are designing for your computer.

This program rolls two dice and lets you see the results, as with real dice. This is especially useful in those games where it is important to see the value of each.

The subroutine in lines 100-140 generates the necessary pair of random numbers. Lines 60, 70 and 80 make the display you want.

Note that lines 60 and 80 each have nine asterisks. Line 150 is RETURN and must be the last line in the program.

After you type in and RUN the program, press any key on your computer's keyboard to roll the dice.

Program Listing

```
10 CLS
20 PRINT"TO ROLL 2 DICE, PRESS ANY KEY"
30 IF INKEY$="" THEN GOTO 30
40 CLS
50 GOSUB 100
60 PRINT"*********"
70 PRINT"* ";L;" * ";R;" *"
```

```
 80 PRINT"*********"
 90 PRINT
 95 GOTO 20
100 RAND
110 LET L=INT(7*RND)
120 IF L<1 THEN GOTO 110
130 LET R=INT(7*RND)
140 IF R<1 THEN GOTO 130
150 RETURN
```

4 See Four Dice

Two dice not enough for your game? Here's how to see four dice after a roll!

Naturally, this program works just like the program in tip number 3 except that the FOR/NEXT loop in lines 50-90 makes the computer roll and display four times rather than two times. If you need six, eight or ten dice on display, change the number two in line 50 to three, four or five.

Program Listing

```
 10 CLS
 20 PRINT"TO ROLL 4 DICE, PRESS ANY KEY"
 30 IF INKEY$="" THEN GOTO 30
 40 CLS
 50 FOR A=1 TO 2
 55 GOSUB 100
 60 PRINT"*********"
 70 PRINT"* ";L;" * ";R;" *"
 80 PRINT"*********"
 85 PRINT
 90 NEXT A
 95 GOTO 20
100 RAND
110 LET L=INT(7*RND)
120 IF L<1 THEN GOTO 110
130 LET R=INT(7*RND)
```

```
140 IF R<1 THEN GOTO 130
150 RETURN
```

5 Secret Message I

Secret messages can be lots of fun! They often are composed of codes in which letters of the alphabet have been replaced by numbers.

In this easy-to-use program, the computer generates a list of pseudorandom numbers and assigns one number to each letter of the alphabet. You use the numbers, in lieu of letters, to write notes to your friends.

There is very little chance of the same number being assigned to two different letters because available numbers range from zero to 999.

By the way, note the nice two-column screen printing format! Line 50 does that.

Program Listing

```
10 RAND
20 LET A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
30 FOR N=1 TO 26
40 LET C=INT(1000*RND)
50 PRINT A$(N);" ";C,
60 NEXT N
```

6 Secret Message II

Now here's something really different in a secret message. The computer asks you for your message, removes it from the screen, stores it, and then displays it backwards!

Yes, we said SDRAWKCAB. It's like looking through your TV tube from behind.

Make up your messages as whole sentences. For instance: I WENT TO THE STORE comes out EROTS EHT OT TNEW I.

Program Listing

```
10 CLS
20 PRINT "WHAT IS THE MESSAGE ?"
30 INPUT A$
40 LET L=LEN A$
50 FOR J=L TO 1 STEP -1
60 PRINT A$(J);
70 NEXT J
80 PRINT
90 GOTO 20
```

7 Up, Down, Back, Forth

"Good golly, what can't that computer do?", will be the question from your surprised neighbor when you show him this neat trick.

You type in any word. The computer instantly prints it on the video display, both up and down vertically, and backward and forward horizontally. It's great to show how smart your computer is when it comes to spelling!

Program Listing

```
 10 PRINT "GIVE ME A WORD"
 20 INPUT A$
 30 CLS
 40 LET L=LEN A$
 50 PRINT "DOWN:","UP:"
 60 FOR J=1 TO L
 70 PRINT A$(J),A$(L+1-J)
 80 NEXT J
 90 PRINT
100 PRINT "FORWARD:"
110 PRINT A$
120 PRINT
```

```
130 PRINT "BACKWARD:"
140 FOR J=L TO 1 STEP -1
150 PRINT A$(J);
160 NEXT J
170 IF INKEY$="" THEN GOTO 170
180 CLS
190 GOTO 10
```

8 Drawing Sketches

Now you can draw lines, rules, diagrams, maps, charts, boxes—anything you can imagine—on the face of your TV set. Use the Computer keyboard as your pen and its video output as your ink.

Lines 120 to 150 accept your up, down, right, or left commands, as U, D, R, or L. No other letters will work. Line 200 draws your lines.

Program Listing

```
 10 CLS
 20 LET X=32
 30 LET Y=22
 40 PLOT X,Y
100 LET A$=INKEY$
110 IF A$="" THEN GOTO 100
120 IF A$="U" THEN LET Y=Y+1
130 IF A$="D" THEN LET Y=Y-1
140 IF A$="R" THEN LET X=X+1
150 IF A$="L" THEN LET X=X-1
200 PLOT X,Y
210 LET A$=""
220 GOTO 100
```

9 Mystery Clues

Want to create your own murder mystery? Figure out whodunit and write your program backwards from there. When your players make wrong guesses, give them tantalizing clues.

Here's a short program which you can load into your computer in a matter of minutes. Key it in and try it out. It shows how you can add clues to your mysteries.

For simplicity, we assume here the Butler did it. Note that, in line 20, we are making him equal to X$. At line 30, the computer stops to ask you whom you think did it. Your answer is recorded in A$.

In line 40, your answer, lodged in A$, is compared with the computer's already-certain knowledge that the Butler did it. A$ is compared with X$. If they agree, and only if they agree, the computer displays the message, "You guessed it." If you got it right, things will end right there.

If, however, you missed it, program execution (sorry about using that word in a murder mystery!) drops to line 50 where we hear the computer, "Clue: servant." After deftly dropping that clue, the computer moves back to line 10 and runs through the whole affair another time. It will keep running through it until you answer, "Butler," in response to its question in line 30.

Program Listing

```
 10 CLS
 20 LET X$="BUTLER"
 30 PRINT "WHODUNIT ?"
 35 INPUT A$
 40 IF X$=A$ THEN GOTO 100
 45 CLS
 50 PRINT "CLUE: SERVANT"
 60 FOR T=1 TO 100
 70 NEXT T
 80 GOTO 10
100 PRINT "YOU GUESSED IT: BUTLER"
```

```
110 PRINT "PRESS ANY KEY TO PLAY AGAIN"
120 IF INKEY$="" THEN GOTO 120
130 GOTO 10
```

10 Original Hi/Lo Game

Here it is. Where everybody started in microcomputer programming back in the Seventies. The first game ever played was a high-low guess-the-number routine.

The computer selects a secret number. You try to guess it. The computer tells you whether or not you are too high, too low, or right on the number.

Here's how it works: the secret number can be zero to 999. Line 100 generates a random number (the secret number) and stores it. Line 210 asks you to guess the number.

Lines 300-310 decide if you are right or wrong. Line 230 keeps track of the number of attempts.

Program Listing

```
 10 LET T=Ø
 20 RAND
100 LET R=INT(1000*RND)
200 PRINT "GUESS THE NUMBER"
210 INPUT B
220 CLS
230 LET T=T+1
240 PRINT "THAT WAS TRY NUMBER ";T
300 IF B>R THEN GOTO 400
310 IF B<R THEN GOTO 500
320 IF B=R THEN GOTO 600
400 PRINT B;" IS TOO HIGH. GUESS AGAIN."
410 PRINT
420 GOTO 200
500 PRINT B;" IS TOO LOW. GUESS AGAIN."
510 PRINT
520 GOTO 200
```

```
600 PRINT
610 PRINT "******************"
620 PRINT R;" IS THE NUMBER"
630 PRINT "******************"
640 PRINT "YOU GOT IT IN ";T;" TRIES"
700 IF INKEY$="" THEN GOTO 700
710 CLEAR
720 CLS
730 GOTO 10
```

11 Code Groups

Need some secret codes for your latest sensitive mission? How about sets of five random letters for use in Morse code practice?

This program has the computer generate an endless string of random combinations of five letters. It won't stop until you press the BREAK key.

Program Listing

```
 10 RAND
 20 LET A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
 30 FOR L=1 TO 5
 40 LET R=INT(100*RND)
 50 IF R<1 OR R>26 THEN GOTO 40
 60 PRINT A$(R);
 70 NEXT L
 80 PRINT
 90 SCROLL
100 GOTO 30
```

Sample Run

CYGQH	IDPLG	MWHOJ
XMIAJ	CTWRQ	KAFDH
BZPDO	VFEZK	BTPNC
REMSF	HALVN	QDFGN
NTVEA	NXECR	ANBWO

12 60-Second Timer

A one-minute timer can be very handy for fun-n-games. This easy-to-use clock counts off seconds up to 60.

The number of seconds counted can be changed by changing the number 60 in line 40.

The clock can be calibrated by changing the number 6 in line 60. Line 60 is a time-delay loop set for approximately one second.

Program Listing

```
 10 CLS
 20 PRINT
 30 SCROLL
 40 FOR T=1 TO 60
 50 PRINT T;" SECONDS"
 60 FOR L=1 TO 6
 70 SCROLL
 80 NEXT L
 90 NEXT T
100 PRINT "END OF TIME"
```

13 Find Highest/Lowest

Suppose we have a list of people and each person has been assigned a number or score. This program accepts the scores and sorts out the persons with the highest and lowest scores. Here's how it works.

Lines 30-130 take in the info on each person. As each person's score is entered, lines 100-120 determine if it is higher or lower than all previous scores. If higher or lower, it is so noted.

To complete data entry, simply press ENTER without data. That will prompt the computer, at lines 140 and 150, to print the lowest score and the highest score.

Program Listing

```
 10 LET N=Ø
 20 PRINT "SCORE:"
 30 INPUT K$
 40 IF K$="" THEN GOTO 130
 50 PRINT K$
 60 LET N=N+1
 70 LET K=VAL K$
 80 IF N=1 THEN LET L=K
 90 IF N=1 THEN LET H=K
100 IF K>H THEN LET H=K
110 IF K<L THEN LET L=K
120 GOTO 20
130 PRINT
140 PRINT "LOWEST SCORE: ",L
150 PRINT "HIGHEST SCORE:",H
160 IF INKEY$="" THEN GOTO 160
170 CLS
180 GOTO 10
```

14 Door Opener

Any good scene needs a door. This program lets you have one to open and close. Here's how it works:

Lines 100 to 140 draw the door. Press C to close the door. Press O to open the door. Lines 300 to 390 draw the doorway outline. Lines 200-240 use INKEY$ to allow you to open or close the door.

Program Listing

```
 10 CLS
100 FOR X=20 TO 25
110 FOR Y=20 TO 30
120 PLOT X,Y
130 NEXT Y
140 NEXT X
200 LET A$=INKEY$
```

```
210 IF A$="" THEN GOTO 200
220 IF A$="O" THEN GOTO 300
230 IF A$="C" THEN GOTO 100
240 GOTO 200
300 FOR X=21 TO 24
310 FOR Y=21 TO 29
320 UNPLOT X,Y
330 NEXT Y
340 NEXT X
350 FOR Y=21 TO 29
360 FOR X=21 TO 24
370 UNPLOT X,Y
380 NEXT X
390 NEXT Y
400 GOTO 200
```

15 Keeping Game Scores

Writing a computer football game? Spelling bee? Cave adventure? No matter what kind of fun you are preparing, you'll need a way to keep score. Here's how.

The wealthy English duke has just been killed in our little mystery game. In lines 10 through 160 of our program listing, below, you play the game, attempting to find out whodunit.

The trick here is in the scorekeeping. Note line 170. If you guessed correctly in response to the query in line 160, at line 170 the computer will give you credit by adding one point to your score stored in memory location R. It does that by comparing your line 160 answer stored in P$ with the correct answer stored in A$.

If you blew it and guessed wrong, the program drops below line 170 to line 180 where it increases your "wrong score" by adding one point to W.

If you got a W+1 at line 190, the program moves back to line 120 and gets you to try again. If you scored a victory and got an R+1 at line 170, the program jumps to line 200 where it stops to display your total right and wrong score.

After that, it's back to line 10 for a complete new run-through.

Program Listing

```
 10 CLEAR
 15 LET W=Ø
 20 LET R=Ø
 25 DIM A$(6,7)
 30 DIM P$(7)
 35 LET S=INT(10*RND)
 40 IF S<1 OR S>6 THEN GOTO 30
 60 LET A$(1)="BUTLER"
 70 LET A$(2)="NANNY"
 80 LET A$(3)="MAID"
 90 LET A$(4)="SON"
100 LET A$(5)="COACH"
110 LET A$(6)="WIFE"
120 PRINT "WHO KILLED THE DUKE?"
130 PRINT "WAS IT THE..."
140 FOR L=1 TO 6
145 PRINT A$(L)
150 NEXT L
160 INPUT P$
165 CLS
170 IF A$(S)=P$ THEN GOTO 200
180 PRINT "NOT ";P$
190 LET W=W+1
195 GOTO 120
200 LET R=R+1
210 PRINT P$;" DID IT"
220 PRINT "YOUR SCORE IS..."
230 PRINT R;" RIGHT ";W;" WRONG"
240 PRINT
250 GOTO 10
```

16 Batting Average

Once you know the number of times you were right and wrong in a game, as in Tip Number 15, it's fun to convert those raw numbers to a batting average. Numbers right and numbers wrong take on a new meaning when changed to a batting average. Folks seem to be able to understand a batting average better.

Our program, starting at line 900, is a partial listing designed to be tacked onto the end of your longer game program to display the final results of play. It will show the number of tries, number of right answers, percentage right, and batting average.

You'll want to test load this program so add lines 10 and 800 as shown. Line 800 will give you the R and T values you'll need going into the program at line 900.

Program Listing

```
 10 CLS
800 LET R=55
810 LET T=100
900 PRINT R;" RIGHT"
910 PRINT "IN ";T;" TRIES"
920 LET D=R/T
930 LET P=100*D
940 LET B=10*P
950 PRINT "THAT IS ";P;" PERCENT"
960 PRINT "BATTING ";B
```

17 Computer Rating Service

Of course, once you know a player's batting average it still might need some interpretation. In this program, the computer takes a look at a batting average and makes a comment.

Remember that this listing, starting here with line 800, is a partial program to be tacked on the end of a longer game. Note that, at 800, you already have values for G (number right) and E (number of tries). Line 810 converts those raw numbers to a batting average (H).

Then, the computer takes that batting average, stored in H, and compares it with values shown in lines

830 to 870. Depending upon the value of H, a slogan is selected by a jump to one of the lines 880 to 950.

By the way, check line 880. You'll see a special epitaph for players with batting averages above 900.

Program Listing

```
 10 CLS
700 LET G=55
710 LET E=100
800 PRINT "YOU GOT ";G;" RIGHT IN "
    ;E;" TRIES"
810 LET H=INT(1000*(G/E))
820 PRINT "BATTING ";H
825 PRINT "YOU ARE..."
830 IF H<100 THEN GOTO 910
840 IF H<300 THEN GOTO 920
850 IF H<500 THEN GOTO 930
860 IF H<700 THEN GOTO 940
870 IF H<900 THEN GOTO 950
880 PRINT "QUALIFIED FOR THE HALL OF FAME"
890 GOTO 960
910 PRINT "VERY NEAR THE BOTTOM OF THE
    BARREL"
915 GOTO 960
920 PRINT "POOR"
925 GOTO 960
930 PRINT "AVERAGE"
940 PRINT "TOP NOTCH"
945 GOTO 960
950 PRINT "DAMN NEAR PERFECT"
960 PRINT "YOUR BATTING AVERAGE IS ";H
```

18 Box Score

To dress up scores during and at the end of a game program, use this method of putting those scores in a box. The box around the score will highlight it and jazz up your video display.

The program here has a temporary substitute for lines 10-40. Normally, you would obtain player's name and score from some larger game program you already have on hand, or are writing. Line 20 gets from you a name and stores it in N$. Line 30 gets a score and stores it in S. If lines 50-230 were a subroutine to a larger program, you would need a RETURN at line 110.

To try the highlighting technique, type in this program just as it is here. All of it, from line 10 through line 230, and RUN it. You'll see the name and score you give the TIMEX displayed in a box on the video screen.

Program Listing

```
 10 DIM A$(10)
 20 PRINT "PLAYERS NAME ?"
 25 INPUT N$
 30 PRINT "PLAYERS SCORE ?"
 35 INPUT S
 40 CLS
 50 LET S$=STR$(S)
 60 LET L=LEN(N$)+LEN(S$)
 70 LET T=L+13
 80 GOSUB 200
 85 PRINT
 90 PRINT "* ";N$;"S SCORE: ";S$;" *"
100 GOSUB 200
110 PRINT
120 PRINT
130 GOTO 20
200 FOR L=1 TO T
210 PRINT "*";
220 NEXT L
230 RETURN
```

Text On Text

19 Create a Quiz

One of the most fascinating uses for your TIMEX is in having it carry on a video conversation with your friends, relatives and neighbors. One useful way to promote such conversation is through a quiz. An instructional, educational quiz, such as we have here.

Quiz data—the computer's storehouse of knowledge—is in lines 100 to 140. Be careful, when you type them in. Spelling and spacing must be exact.

Of course, the quiz can be made much longer. In this example, it could be expanded to encompass all past U.S. presidents.

Program Listing

```
 10 RAND
 20 DIM P$(5,10)
100 LET P$(1)="WASHINGTON"
110 LET P$(2)="ADAMS"
120 LET P$(3)="JEFFERSON"
130 LET P$(4)="MADISON"
140 LET P$(5)="MONROE"
200 LET R=INT(6*RND)
210 IF R<1 THEN GOTO 200
300 PRINT "WHO WAS PRESIDENT NO. ";R
310 PRINT "OF THE UNITED STATES ?"
320 INPUT A$
330 IF A$="" THEN GOTO 320
340 LET L=LEN A$
350 CLS
360 IF A$=P$(R,1 TO L) THEN GOTO 500
400 PRINT "WRONG"
410 GOTO 510
500 PRINT "CORRECT"
510 PRINT P$(R);" WAS PRESIDENT"
520 PRINT "NUMBER ";R
530 PRINT
540 GOTO 10
```

20 Killing Time

Sometimes, it may seem to you as if the computer will never get to the result of a job. You understand the processing delay but your non-computer friends may not. They could be confused by the wait and think the computer is "broken."

To keep their minds off the slowness, give them something to look at while the computer is "thinking."

The added, extra lines, numbered 60 and 70, take up more processing time but make for less confusion. Computing may take a bit longer but your fun will be increased.

If you delete lines 60-70 you'll see how the program runs faster but the blank screen is confusing.

Program Listing

```
 10 LET X=Ø
 20 PRINT "GIVE ME A NUMBER"
 30 INPUT N
 40 FOR L=1 TO N
 50 LET X=X+L
 60 CLS
 70 PRINT "I AM THINKING"
 80 NEXT L
 90 CLS
100 PRINT "I HAVE THE ANSWER"
110 PRINT "THE TOTAL OF ALL NUMBERS"
120 PRINT "FROM 1 TO ";N;" IS ";X
130 PRINT
140 GOTO 10
```

21 Word-Error Trapping

The same kind of error trapping is available for strings. Suppose the program, as in this example, asks at line 10 for a word. It is looking for YES or NO. If it gets a YES, then line 20 sees that it got what it wanted and moves operations along to line 100.

If it gets a NO, then line 20 hasn't received what it wants so program execution moves on to line 30. Here, at line 30, the program finds something useful and shoots operations down to line 200.

If, however, neither YES nor NO were entered at line 10, then neither lines 20 nor 30 would be satisfied so action would drop to line 40. Here, the error is trapped by commanding the operator to give one of the two correct answers. Then, at line 50, the operation is returned to line 10 for a new try at the correct input.

Program Listing

```
  5 PRINT "WANT TO PLAY AGAIN ?"
 10 INPUT A$
 20 IF A$="YES" THEN GOTO 100
 30 IF A$="NO" THEN GOTO 200
 40 PRINT "PLEASE ANSWER ONLY YES OR NO"
 50 GOTO 10
100 PRINT A$
110 STOP
200 PRINT A$
210 STOP
```

22 Character Numbers

This brief program displays the value for each keyboard character, side-by-side with the character it stands for. You will be able quickly to tell what each number prints.

Line 40 is a timing loop to slow down the presentation so you can digest the information. To make it even slower, increase the number in line 40. To make it faster, decrease the number in line 40.

Program Listing

```
10 FOR L=Ø TO 255 STEP 21
20 FOR N=L TO L+21
30 PRINT N,CHR$ N
40 PAUSE 60
50 NEXT N
60 CLS
70 NEXT L
```

23 One-Time Password

If you don't want unauthorized use of your programs, insert a requirement that a user know a password. This particular routine allows only one try at entering a correct password.

For our password, we have selected "elephant" and stored it in line 30. You can change the password to whatever you like.

If a correct attempt at entering the password is made, program action will progress to line 100. Otherwise, action drops to line 40 and action ends.

Program Listing

```
 10 PRINT "WHAT IS THE PASSWORD ?"
 20 INPUT A$
 30 IF A$="ELEPHANT" THEN GOTO 100
 40 STOP
100 PRINT "YOU GOT IT RIGHT"
110 PRINT "A PROGRAM WOULD RUN"
```

24 Three-Tries Password

Here the software lets you try three times to enter the correct password. You don't get to go forward with the program if you don't get it right in three tries.

Again the password is "elephant" and is stored in line 30. You can change the password to whatever suits you.

Lines 40 to 60 allow the three attempts. If no good after three tries, then end.

Program Listing

```
  5 LET B=Ø
 10 PRINT "WHAT IS THE PASSWORD ?"
 20 INPUT A$
 30 IF A$="ELEPHANT" THEN GOTO 100
 40 LET B=B+1
 50 IF B=3 THEN STOP
 60 GOTO 10
100 PRINT "YOU GOT IT RIGHT"
110 PRINT "A PROGRAM WOULD RUN"
```

25 Multiple Passwords

Here's a really complex password entry system. It has a unique "account number" and a password for each person. This will allow several different persons access to the program but each person will have a different combination to the lock!

account number	password
12345	zebra
23456	goose
34567	trout
45678	snake

Each individual user must correctly enter his unique account number and then his own personal password. If account number is wrong, then the password never can be right. If account number is okay but password doesn't match, the user gets no run.

You can add users to this program by adding lines to the 300-340 subroutine.

Program Listing

```
 10 FOR L=1 TO 3
 20 PRINT "YOUR ACCOUNT NUMBER ?"
 30 INPUT U
 40 GOSUB 300
 50 PRINT "PASSWORD ?"
 60 INPUT P$
 70 IF P$=W$ THEN GOTO 100
 80 NEXT L
 90 STOP
100 PRINT "YOU GOT IT RIGHT"
110 PRINT "A PROGRAM WOULD RUN"
120 STOP
300 IF U=12345 THEN LET W$="ZEBRA"
310 IF U=23456 THEN LET W$="GOOSE"
320 IF U=34567 THEN LET W$="TROUT"
330 IF U=45678 THEN LET W$="SNAKE"
340 RETURN
```

26 Memory Review

This useful program examines a range of memory locations you specify and reports what it finds there. It displays the memory location number alongside the contents, for convenience.

Program Listing

```
 10 PRINT "SEARCH ROM MEMORY"
 20 PRINT "FOR CONTENTS FROM..."
 30 PRINT "LOW ADDRESS: ";
 40 INPUT L
 50 PRINT L
 60 PRINT "HIGH ADDRESS: ";
 70 INPUT H
 80 PRINT H
```

```
 90 PRINT
100 PRINT "ADDRESS";TAB 10;"CONTENTS"
    ;TAB 20;"CHARACTER"
110 FOR X=L TO H
120 LET C=PEEK X
130 PRINT X;TAB 10;C;TAB 20;CHR$ C
140 NEXT X
```

27 Entering: Letter Stop

One way to conclude an input series, and get out of its entry loop, is to use a key letter to promote a jump. In this brief example, we input numbers, at line 100, as string values. If we give the computer an X rather than a number, it will jump down to line 200 for new action.

Numbers keyed in are stored first as strings. Then line 120 changes them to number values for the addition in line 130.

Program Listing

```
 10 LET B=Ø
 20 LET C=Ø
 30 PRINT "GIVE ME A NUMBER: ";
100 INPUT A$
110 IF A$="X" THEN GOTO 200
115 PRINT A$
120 LET B=VAL A$
130 LET C=C+B
140 GOTO 30
200 PRINT
210 PRINT
220 PRINT "TOTAL OF NUMBERS = ";C
```

28 Entering: Zero Stop

Here's another way to conclude an entry loop: have the computer be on the lookout for a plain zero. When a zero is entered, the computer will jump out of the entry cycle and on to further action.

This program totals numbers as they are added and accumulates them in memory location B. If one of the numbers entered is a zero alone, then line 110 will spot it and send the computer on down to line 200, breaking the entry cycle.

Naturally, you can't use a zero in a string of numbers to be added since zero causes the computer to quit entering and get on with displaying.

Program Listing

```
 10 LET B=Ø
 20 PRINT "GIVE ME A NUMBER: ";
100 INPUT A
110 IF A=Ø THEN GOTO 200
120 PRINT A
130 LET B=B+A
140 GOTO 20
200 PRINT
210 PRINT
220 PRINT "TOTAL OF NUMBERS = ";B
```

29 Superior Decision Maker

Remember that YES/NO Executive Decision Maker which was so popular? In this superior edition, a choice of eight replies is possible.

Program Listing

```
 10 DIM D$(8,16)
 20 RAND
100 LET D$(1)="YES"
110 LET D$(2)="FIRE SOMEONE"
120 LET D$(3)="PASS THE BUCK"
130 LET D$(4)="MAYBE"
140 LET D$(5)="REORGANIZE"
150 LET D$(6)="NO"
```

```
160 LET D$(7)="SEE YOUR ANALYST"
170 LET D$(8)="SIT ON IT"
200 LET R=INT(9*RND)
210 IR R<1 THEN GOTO 200
300 PRINT D$(R)
400 IF INKEY$="" THEN GOTO 400
410 CLS
420 GOTO 10
```

30 Stopwatch

Now you can leave that chrome-plated stopwatch at home next time you travel to your favorite auto or horse race. This program turns your TIMEX into a handy stopwatch timer using the TV display.

When you RUN the program, the stopwatch will start counting seconds.

You can adjust the accuracy of the seconds count by changing the wait number in line 110. We show it set at 10. To show down the timer, increase that number. To speed up the clock, decrease the number.

Program Listing

```
 10 PRINT "NUMBER OF SECONDS"
 20 PRINT "TO BE COUNTED DOWN: ";
 30 INPUT S
 40 PRINT S
 50 PRINT
 60 PRINT "PRESS ANY KEY"
 70 PRINT "TO START TIMING"
 80 IF INKEY$="" THEN GOTO 80
 90 CLS
100 FOR L=S TO 1 STEP -1
110 PAUSE 10
120 PRINT L
130 NEXT L
140 CLS
150 PRINT "TIME IS UP"
```

```
160 PRINT S;" SECONDS HAVE ELAPSED"
170 PRINT
180 PRINT
190 GOTO 10
```

31 Wipeout!

Warning: handle with care!

Careless operation of this program can cause you a lot of extra work.

Key in the program. Run it. When it asks for the password, be sure to give it what it wants or it will erase itself. That's right, the entire contents of program memory down the tubes!

Here's how it works:

The password, in this case Tracey , is asked by line 20. You give it a password. In line 30, your answer is compared to the true password. If correct, action goes to line 50. If incorrect, the program goes to line 40.

The NEW statement in line 40 erases everything from program memory.

You can change the password in line 30 to any letters, numbers or keyboard symbols of your choice. Watch out when testing. A wrong password can cause a lot of retyping.

Program Listing

```
10 PRINT "WHAT IS THE SECRET WORD ?"
20 INPUT W$
30 IF W$="TRACEY" THEN GOTO 50
40 NEW
50 PRINT "CORRECT"
60 PRINT "THE SECRET WORD IS ";W$
```

32 Sentence Writer

Practice your English!

Exhibit your knowledge of nouns and verbs. This program leads the computer to solicit individual words from you and use those words to create sentences.

Besides helping you better understand verbs, nouns and simple declarative sentence structures, the program demonstrates the computer's ability to simulate conversation and communication.

Lines 20, 30 and 40 take in the words. Then action goes on to line 70 where the program pushes the computer back up to line 10. At line 10, everything starts over.

You may modify the program to suit your own interests or needs.

Program Listing

```
10 PRINT "A PLURAL NOUN = ?";
20 INPUT N$
25 PRINT "A VERB = ?";
30 INPUT V$
35 PRINT "A SINGULAR NOUN = ?";
40 INPUT S$
50 CLS
60 PRINT "THE "+N$+" "+V$+" "+S$+"."
65 PRINT
70 GOTO 10
```

Sample Run

```
A PLURAL NOUN = ?
DOGS
A VERB = ?
LOVE
A SINGULAR NOUN = ?
FOOD

THE DOGS LOVE FOOD.
```

```
A PLURAL NOUN = ?
BOXES
A VERB = ?
HOLD
A SINGULAR NOUN = ?
WATER

THE BOXES HOLD WATER
```

33 Categorizing

A large quantity of numbers can be categorized and thereby cut down into a smaller quantity of numbers. See our example: it takes test scores and divides them into ranges labeled A, B, C, D, and F.

The program assumes exam or test scores in a range of zero to 100. The letter grades include zero to 59, F; 60-69, D; 71-79, C; 80-89, B; 90-100, A.

Key in as many scores as you like and then enter the letter X to stop the entry cycle.

Lines 100-140 sort all scores into the A through F categories. Lines 150-170 sort highest and lowest scores.

Program Listing

```
 10 GOSUB 400
 20 PRINT "SCORE="
 30 INPUT G$
 40 CLS
 50 IF G$="X" THEN GOTO 200
 60 LET G=VAL G$
 70 LET N=N+1
100 IF G<60 THEN LET F=F+1
110 IF G>59 AND G<70 THEN LET D=D+1
120 IF G>69 AND G<80 THEN LET C=C+1
130 IF G>79 AND G<90 THEN LET B=B+1
140 IF G>89 THEN LET A=A+1
150 IF N=1 THEN LET L=G
155 IF N=1 THEN LET H=G
```

```
160 IF G<L THEN LET L=G
170 IF G>H THEN LET H=G
180 GOTO 20
200 PRINT "TOTAL ";N;" SCORES"
210 PRINT "FROM ";L;" TO ";H
220 PRINT "A",A
230 PRINT "B",B
240 PRINT "C",C
250 PRINT "D",D
260 PRINT "F",F
270 STOP
400 CLEAR
410 LET A=Ø
420 LET B=Ø
430 LET C=Ø
440 LET D=Ø
450 LET F=Ø
460 LET N=Ø
470 RETURN
```

34 Alphabet Soup

Sure, everybody knows there are 26 letters in the alphabet. But, do you know which letter is number 20? Number 5? Number 17? Well, your Computer knows!

Type in this short ready-to-run program. RUN it. The computer will spit out number-and-letter combinations all day long. The number on the left is the position in the alphabet of the letter on the right.

It's a fun way to demonstrate to your friends just how "smart" the computer is!

How does it work? The alphabet is stored, at line 20. Line 30 generates a random number. Line 40 cuts off any zeros which are generated. Now the useful random numbers range from one to 26.

Line 60 then uses the current random number, from one to 26, to pick one letter from the character variable Z$.

Line 90 causes action to jump back to line 10 where the whole process starts over.

Program Listing

```
10 RAND
20 LET Z$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
30 LET P=INT(27*RND)
40 IF P<1 THEN GOTO 30
60 PRINT P,Z$(P)
80 SCROLL
90 GOTO 10
```

Sample Run

```
21  U   |  17  Q   |  19  S
 3  C   |  11  K   |  18  R
25  Y   |   8  H   |  14  N
 6  F   |  10  J   |  20  T
 2  B   |   7  G   |   5  E
        |          |  19  S
```

35 Letter Repeaterrrr

The screen remains blank until you press any key. Then, the computer prints and repeats printing the key you pressed until a different key is pressed.

Printing and repetition of the new character starts where the previous character ended on the screen.

The 26 letters of the alphabet can be printed and repeated on the screen.

INKEY$ makes it possible for the computer to know which keyboard character you have pressed. The problem is instructing the computer to hold and reprint that key's character while watching for any new key press. We solve that problem here by the test in line 20. The test has the computer look to see if anything is stored in A$.

If a key has been pressed, INKEY$ in line 10 will deposit that information in A$.

Program Listing

```
10 LET A$=INKEY$
20 IF A$="" THEN GOTO 10
```

```
30 PRINT A$;
40 GOTO 10
```

36 Question & Answer

Here's how to use the DATA statement, and the computer's ability to, search for data, to create a Q&A.

How many days are there in a month? It's a tough question for grade schoolers and some extra study may be in order. Use this program.

The program has the computer present the number of a month and then ask how many days are in that particular month.

If you enter a correct answer, the computer will say so. If you enter an incorrect answer, the computer will say your anwer was wrong and tell the correct answer.

The program runs on forever until you press the BREAK key.

Program Listing

```
 10 DIM D(12)
 20 RAND
100 LET D(1)=31
110 LET D(2)=28
120 LET D(3)=31
130 LET D(4)=30
140 LET D(5)=31
150 LET D(6)=30
160 LET D(7)=31
170 LET D(8)=31
180 LET D(9)=30
190 LET D(10)=31
200 LET D(11)=30
210 LET D(12)=31
300 LET R=INT(13*RND)
310 IF R<1 THEN GOTO 300
400 PRINT" HOW MANY DAYS IN MONTH ";R
```

```
410 INPUT D
420 CLS
430 IF D=D(R) THEN GOTO 600
500 PRINT "WRONG"
510 GOTO 610
600 PRINT "CORRECT"
610 PRINT "MONTH ";R;" HAS ";D(R);" DAYS"
620 PRINT
630 GOTO 10
```

Gee Whiz

37 Gee Whiz I: Smart Adder

These six programs, in this section of the book, make up our *Gee Whiz* series. One of the fun ways to use your TIMEX is in wowing your friends. Next time they ask, "But, what can it do?", show them its uncanny abilities at adding, spelling, writing upside down, even cracking jokes. Try these six *Gee Whiz* programs on your friends. You'll love their reactions.

Smart Adder is the first in the series. When your neighbor drops in for a cup of coffee, bring out the TIMEX for a demonstration of its lightning speed.

This program adds long strings of numbers in a flash. You give the computer a number. It starts at 1 and adds all numbers up to and including your number. For instance, if you give it a five, it will add 1 plus 2 plus 3 plus 4 plus 5 and display the result.

Ask your neighbor how fast he or she can add all the numbers to 100. It should take several minutes. While he's working on it, let your TIMEX do it in a split second. Your neighbor's reaction is bound to be, "Gee whiz!"

Program Listing

```
 10 LET X=Ø
 20 PRINT "GIVE ME A NUMBER"
 30 INPUT N
 40 FOR L=1 TO N
 50 LET X=X+L
 60 NEXT L
 70 PRINT
 80 PRINT "THE TOTAL OF 1 TO ";N;" IS "
    ;X
 90 PRINT
100 CLEAR
110 GOTO 10
```

38 Gee Whiz II: Three-Digit Mystery

Have your neighbor secretly select any three-digit number in which all three digits are the same. Then have him tell the computer only the *sum* of those three digits.

The computer will identify his secret number!

Program Listing

```
 10 CLS
 20 PRINT "SELECT A THREE-DIGIT NUMBER"
 30 PRINT "WITH ALL THREE DIGITS THE
    SAME"
 40 PRINT
 50 PRINT "ADD THE THREE DIGITS TOGETHER"
 60 PRINT "WHAT IS THE SUM"
 65 PRINT "OF THE THREE DIGITS ?"
 70 INPUT N
 75 CLS
 80 LET Q=37*N
 90 PRINT
100 PRINT "YOUR ORIGINAL NUMBER WAS ";Q
110 PRINT
120 PRINT
130 GOTO 20
```

39 Gee Whiz III: Yes/No Decision Maker

This is handy for the busy executive who doesn't have time for decisions.

Line 10 clears the screen. Line 20 generates a random number from zero to 99. Line 30 selects a *yes* answer if the random number is greater than 49. Otherwise, line 40 chooses a *no* answer.

Program Listing

```
10 CLS
20 LET X=100*RND
30 IF X>49 THEN PRINT "YES"
40 IF X<49 THEN PRINT "NO"
50 IF INKEY$="" THEN GOTO 50
60 GOTO 10
```

40 Gee Whiz IV: First Alphabet Spotter

There are 26 letters in the alphabet. Each has a number. For instance, number 1 is A. Number 20 is T. This *Gee Whiz* program has the computer ask you for a number from 1 to 26 and then, faster than a jackrabbit, tell you what letter it goes with.

Naturally, you'll know how it works but to your non-computer friends it will seem like the computer is a genius!

Program Listing

```
 10 CLS
 20 PRINT "GIVE ME THE NUMBER OF"
 30 PRINT "A LETTER FROM THE ALPHABET"
 40 PRINT "FROM 1 TO 26"
 50 INPUT N
 60 LET X=N+37
 70 CLS
 80 PRINT "LETTER NUMBER ";N;" IS ";CHR$ X
 90 PRINT
100 GOTO 20
```

41 Gee Whiz V: Second Alphabet Spotter

This *Gee Whiz* program has the computer ask you for another number from 1 to 26 and then, faster than a jackrabbit, tell you what letter it goes with.

Program Listing

```
 10 CLS
 20 CLEAR
 30 LET A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
 40 PRINT "GIVE ME THE NUMBER OF"
 50 PRINT "A LETTER FROM THE ALPHABET"
 60 PRINT "FROM 1 TO 26"
 70 INPUT N
 80 CLS
 90 PRINT "LETTER NUMBER ";N;" IS "
    ;A$(N)
100 IF INKEY$="" THEN GOTO 100
110 GOTO 10
```

42 Gee Whiz VI: Guess The Number

Here it is! The world's oldest, longest running, most popular game: Guess The Number.

When you start the program running, the computer thinks of a number and stores that away. You try to guess the number. If your number is too high, the computer says, "TOO HIGH."

If you are too low, the computer will report "TOO LOW." The possible numbers range from zero to 100.

Program Listing

```
 10 CLS
 20 CLEAR
 30 LET Q$="*"
 40 GOTO 130
 50 LET N=INT(101*RND)
 60 PRINT "GUESS THE NUMBER"
 70 INPUT G
 80 IF G>N THEN PRINT "TOO HIGH"
 90 IF G<N THEN PRINT "TOO LOW"
100 IF G<>N THEN GOTO 60
110 PRINT "RIGHT"
120 PRINT "LETS GO AGAIN"
130 FOR L=1 TO 16
140 PRINT Q$;
150 NEXT L
160 PRINT
170 GOTO 50
```

Sample Run

```
********************
GUESS THE NUMBER ?
46
TOO HIGH
GUESS THE NUMBER ?
21
TOO LOW
GUESS THE NUMBER ?
25
RIGHT !
LETS GO AGAIN
********************
```

Number Crunching

43 Memory Tester

Most everybody can remember numbers. At least short numbers with few digits. But how long a number can you recall in a flash?

The computer will briefly display a number. It then will remove the number from your view and ask you to repeat what it was. If you make a mistake in your reply, it won't show you the number again. If you miss three times, the computer will tell you to FORGET IT, give you your score and end the game. Then it will start over.

On the other hand, if you recall correctly, the computer will say so and then give you a new number. The new number will have more digits than the previous number. Each time you guess correctly, the number gets longer.

No matter how good you are, at some point you won't be able to recall *all* the digits in proper sequence.

By the way, if you just can't operate at the speed of this program, you can slow down the display by changing the number 100 in line 65.

How many digits can you quickly recall?

Program Listing

```
 10 LET W=Ø
 15 LET R=Ø
 20 LET Z=1
 30 LET S=INT(100*RND)
 40 LET N=INT(S*Z)
 60 PRINT "REMEMBER ";N
 65 PAUSE 100
 70 CLS
 80 IF W=3 THEN PRINT "FORGET IT"
 85 IF W=3 THEN GOTO 145
 90 PRINT "WHAT WAS IT ?"
 95 INPUT S
100 IF S<>N THEN PRINT "WRONG"
105 IF S<>N THEN LET W=W+1
110 IF S<>N THEN GOTO 80
115 PRINT "RIGHT"
```

```
120 LET R=R+1
125 LET W=Ø
130 LET Z=11*Z
135 PRINT R;" RIGHT SO FAR"
140 GOTO 30
145 PRINT "YOU HAD ";R;" RIGHT"
150 PRINT "LETS START OVER"
155 GOTO 10
```

44 Number Reverser I

This program generates a series of random numbers and then adds them up. Then it reverses the order of the digits in the total. Creates a very strange random number!

Lines 20-40 generate three random single-digit numbers. Line 50 is a trap to make sure they are between 1 and 9 with no zeros showing up.

Line 60 changes the number values into strings so they can be tied together as three digits at line 75. Line 85 takes those three-digit strings from line 75 and turns them back into numerical values.

Line 105 is a time delay loop to stall a moment before line 110 pushes action back up to line 10 for a new start.

Program Listing

```
10 CLS
20 LET J=INT(10*RND)
30 LET K=INT(10*RND)
40 LET L=INT(10*RND)
50 IF J<1 OR K<1 OR L<1 THEN GOTO 20
60 LET J$=STR$ J
65 LET K$=STR$ K
70 LET L$=STR$ L
75 LET N$=J$+K$+L$
80 LET R$=L$+K$+J$
85 LET N=VAL N$
90 LET R=VAL R$
95 PRINT "RANDOM NUMBER IS ";N
```

```
100 PRINT "BACKWARD,  IT IS ";R
105 PAUSE 200
110 GOTO 10
```

45 Number Reverser II

Give the computer a number. It will immediately, display the number printed backward on the video display.

Line 20 collects the number from you. Line 30 measures its length. Lines 40-70 reverse it and line 80 displays the reversed number.

Program Listing

```
 10 LET B$=""
 20 PRINT "GIVE ME A NUMBER ";
 25 INPUT N$
 30 LET L=LEN N$
 35 PRINT N$
 40 FOR Y=L TO 1 STEP -1
 50 LET B$=B$+N$(Y)
 60 NEXT Y
 70 LET B=VAL B$
 80 PRINT "REVERSED, THE NUMBER IS ";B
 90 PRINT
100 GOTO 10
```

46 Number-Error Trapping

Good programs, those which are well written, need *error trapping*. It's a technique for making sure persons communicating with the computer don't key in inappropriate data or make mistakes which would cause computation problems for the computer.

For instance, see the example program here. In line

10 the computer asks for a number. In line 20, if the number is too low, it says so and goes back to line 10 to repeat its request.

At line 30, if the number received at line 10 is too large, it says so and goes back to line 10 for a better choice.

The result is only printed at line 40 when a satisfactory number has been keyed in back at line 10.

You can set your own limits by changing the 10 in line 20 and the 100 in line 30.

Program Listing

```
10 PRINT "GIVE ME A NUMBER"
15 INPUT A
20 IF A<10 THEN PRINT "TOO LOW"
30 IF A>100 THEN PRINT "TOO HIGH"
35 IF A<10 OR A>100 THEN GOTO 10
40 PRINT A;" IS OKAY"
50 PRINT
60 PRINT
70 GOTO 10
```

47 Standard Deviation

Here's a way to determine mean and standard deviation. In this particular program, you exit the entry cycle by entering the large number 999999999 (nine 9's) so you can't use 999999999 as one of your data points.

This is a great opportunity to experiment with standard deviation computations. Try a series of data points such as 3, 5, 3, 7, and 4. They should result in a total of 22; mean of 4.4; variance of 2.24; and standard deviation of 1.496663.

Program Listing

```
10 GOTO 160
20 PRINT "DATA POINT: ";
25 INPUT X
```

```
 30 IF X=999999999 THEN GOTO 55
 33 PRINT X
 35 LET T=T+X
 40 LET S=S+(X*X)
 45 LET N=N+1
 50 GOTO 20
 55 LET A=T/N
 60 LET V=(S/N)-(A*A)
 65 LET D=SQR V
 70 CLS
 80 PRINT "DATA POINTS TOTAL ";T
 90 PRINT "MEAN IS ";A
100 PRINT "VARIANCE IS ";V
110 PRINT "STANDARD DEVIATION IS ";D
120 PRINT
130 PRINT
140 CLEAR
150 IF INKEY$="" THEN GOTO 150
160 LET T=Ø
170 LET S=Ø
180 LET N=Ø
190 LET A=Ø
200 LET V=Ø
210 LET D=Ø
220 GOTO 20
```

48 Percentages

Usually it's more convenient to enter percentages as percent rather than having to convert to decimals in your head first. Of course, the computer needs that converted decimal value to do its work. How to get it?

This program does the trick. You give it a percentage and it converts that to a decimal. The computer does the hard work for you!

Line 30 makes the actual conversion. Use this idea as part of a larger check-balancing, accounting or bookkeeping program and save lots of mental effort.

Program Listing

```
10 PRINT "WHAT IS THE PERCENTAGE ?"
20 INPUT P
30 LET D=0.01*P
40 PRINT "THE DECIMAL IS ";D
50 PRINT
60 CLEAR
70 GOTO 10
```

49 Logic Functions

You can make your computer do things based on its decision that something exists. That is, in the first program listing here, it only will print the value of C if it finds that B has an existing value. If B is found to have no value, does not exist, C will not be printed.

The decision is in line 40. The machine only prints C if B does not equal zero. Since, in line 20, we set B = 10, the computer will find that something exists in B and, thus, go ahead and do the work assigned in the last half of line 40. If nothing had been stored in B, the last half of line 40 would have been ignored.

Program Listing

```
10 CLS
20 LET B=10
30 LET C=10*B
40 IF B THEN PRINT C
```

In the second program here, the TIMEX only displays the results of the tests in lines 40 and 50 if the results of one or both is "true."

By doing the simple math in your head, you can see that the information in the right-hand side of line 20 is true. The information in the right-hand side of line 30 is false.

Line 20 says that 6 + 8 is greater than 3 times 4. That is, 14 is greater than 12. That is true.

Line 30 says that 5 + 2 is greater than 9 + 2. That is, 7 is greater than 11. That is false.

After reading line 20, the computer will store a 1 in B since the statement is true. Upon reading line 30, the computer will store a zero in C since the statement is false.

As action drops to line 40, the computer will find the 1 it stored in B and, thus, complete the action called for at the right-hand end of line 40. It will display the message, "B OKAY."

At line 50, however, the computer will find "nothing" (zero) in C and will not complete the right-hand end of that instruction. It only will do the right-hand end if it finds something in the left-hand end.

These logic functions are great for quick tests.

Program Listing

```
10 CLS
20 LET B=(6+8)>(3*4)
30 LET C=(5+2)>(9+2)
40 IF B THEN PRINT "B OKAY"
50 IF C THEN PRINT "C OKAY"
```

50 Above & Below a Line

Here's a way to count numbers above and below a cut-off line. The computer solicits numbers between 1 and 100. Any numbers you key in which are below 1 or above 100 are trapped out by line 40. Entering a zero ends the input cycle.

Line 50 counts the total numbers. Line 60 counts only those numbers between 1 and 50. Line 80 counts the numbers from 51 to 100. Lines 90 to 130 present results.

Program Listing

```
10 GOTO 160
20 PRINT "GIVE ME A NUMBER ";
```

```
 25 INPUT Z
 30 IF Z=Ø THEN GOTO 80
 35 PRINT Z
 40 IF Z<1 OR Z>100 THEN GOTO 20
 50 LET N=N+1
 60 IF Z<51 THEN LET B=B+1
 70 GOTO 20
 80 LET A=N-B
 90 CLS
100 PRINT "TOTAL NUMBERS: ",N
110 PRINT "1 TO 50: ",B
120 PRINT "51 TO 100: ",A
130 PRINT
160 LET N=Ø
170 LET A=Ø
180 LET B=Ø
190 GOTO 20
```

51 Factoring

This program finds and lists the factors of any number you specify, up to 110. It can be used as a subroutine in a larger program, with appropriate attention to line numbers, variable names, and RETURN.

The number of individual factors are limited by the DIM statement in line 20.

The list will exclude the number itself divided by 1.

For a quick sample run, try the number 18. You should find factors are 9, 6, 3 and 2. Press any key to restart.

Program Listing

```
10 GOTO 160
20 DIM Q(55)
30 PRINT "NUMBER=";
35 INPUT N
40 PRINT N
45 FOR L=2 TO N/2
```

```
 50 LET M=N/L
 60 IF M=INT M THEN LET Q(L)=M
 70 NEXT L
 80 PRINT "FACTORS ARE:"
 90 FOR L=1 TO N/2
100 IF Q(L)>1 THEN PRINT Q(L)
105 IF Q(L)>1 THEN GOTO 120
110 LET Z=Z+1
120 NEXT L
130 IF N=1 THEN PRINT "NONE"
135 GOTO 145
140 IF Z=INT N/2 THEN PRINT "NONE"
150 CLS
160 LET M=Ø
170 LET Z=Ø
180 GOTO 20
```

52 Which is Smallest?

How can the computer tell which number is smaller or larger? Here's how.

Type in the program and RUN it. It will ask for, and accept a continuous string of numbers until you end the input routine by keying in a zero.

Lines 40 to 60 make the decision as to which number is lowest.

Press any key to restart .

Program Listing

```
10 LET N=Ø
15 LET D=Ø
20 PRINT "GIVE ME A NUMBER ";
25 INPUT Z
30 IF Z=Ø THEN GOTO 80
35 PRINT Z
40 LET N=N+1
50 IF N=1 THEN LET D=Z
60 IF Z<D THEN LET D=Z
```

```
 70 GOTO 20
 80 CLS
 90 PRINT "THE SMALLEST NUMBER WAS ";D
100 IF INKEY$="" THEN GOTO 100
110 CLS
120 GOTO 10
```

53 Which is Largest?

Suppose you have a group of numbers and you would like to know which number is largest within the group? Here's a software routine for your TIMEX so it can locate the largest number.

You can key in as many numbers as you wish. To end that entry cycle, type in a zero. The computer will see that zero as its cue to leave the entering routine and get on with computing.

Press any key to restart.

Program Listing

```
 10 LET N=Ø
 15 LET H=Ø
 20 PRINT "GIVE ME A NUMBER ";
 25 INPUT Z
 30 IF Z=Ø THEN GOTO 80
 35 PRINT Z
 40 LET N=N+1
 50 IF N=1 THEN LET H=Z
 60 IF Z>H THEN LET H=Z
 70 GOTO 20
 80 CLS
 90 PRINT "THE LARGEST NUMBER: ";H
100 IF INKEY$="" THEN GOTO 100
110 CLS
120 GOTO 10
```

54 Reciprocals

Key in any number. The computer will display its reciprocal. The actual conversion is done here at line 30.

Program Listing

```
10 LET R=Ø
20 PRINT "NUMBER TO CONVERT TO RECIPROCAL"
25 INPUT N
30 LET R=1/N
35 CLS
40 PRINT "THE RECIPROCAL OF ";N
45 PRINT "IS ";R
50 PRINT
55 GOTO 10
```

55 Dump the Integer

Look at the number 123.456 with an eye toward how to get rid of the portion left of the decimal point. Keep only .456 and dump 123. Here's a short program to accomplish that.

Try 5.67. It will come out .67. Or 500.5 which will come out .5.

Caution: your TIMEX is *not* always accurate at math. Try 987.65, for example. It will result in .64999986 because when the TIMEX subtracts 987 from 987.65 at line 30 it does not get .65 as the correct answer. Rather, it gets an approximate answer!

Program Listing

```
10 LET X=Ø
20 PRINT "GIVE ME ANY NUMBER"
30 PRINT "WITH A DECIMAL"
40 INPUT N
50 LET X=N-INT N
```

```
 60 CLS
 70 PRINT "THE FRACTIONAL PORTION"
 80 PRINT "OF ";N;" IS ";X
 90 PRINT
100 GOTO 10
```

56 Averages

Key in numbers in any order. A zero will end entry. The computer will tell you the average number of all numbers you entered.

Program Listing

```
 10 LET N=Ø
 15 LET T=Ø
 20 LET A=Ø
 25 PRINT "GIVE ME A NUMBER ";
 30 INPUT Z
 35 IF Z=Ø THEN GOTO 70
 40 PRINT Z
 45 LET N=N+1
 50 LET T=T+Z
 60 GOTO 25
 70 LET A=T/N
 80 CLS
 90 PRINT "THE AVERAGE NUMBER IS ";A
100 PRINT
110 GOTO 10
```

57 Mid-Range Number

Here's how to find the middle of a range of numbers. You key in as many numbers in a series as you wish. After the last number, key in a zero to move the program out of the entry cycle.

Lines 40 to 70 select the highest and lowest numbers in the range. They actually define the range. Then line 90 finds the middle point of that range.

Program Listing

```
10 LET N=Ø
15 LET M=Ø
20 LET H=Ø
25 LET L=Ø
30 PRINT "GIVE ME A NUMBER ";
35 INPUT Z
40 IF Z=Ø THN GOTO 90
45 PRINT Z
50 LET N=N+1
55 IF N=1 THEN LET H=Z
60 IF N=1 THEN LET L=Z
65 IF Z<L THEN LET L=Z
70 IF Z>H THEN LET H=Z
80 GOTO 30
90 LET M=L+((H-L)/2)
100 CLS
110 PRINT "THE MID-RANGE NUMBER IS ";M
120 PRINT
130 GOTO 10
```

58 Rounding Off

The technique for rounding off numbers is easy. This program, which can stand alone or be worked into a larger program as a subroutine, rounds a decimal to the nearest whole number.

Program Listing

```
10 LET R=Ø
15 LET D=Ø
20 PRINT "GIVE ME A NUMBER"
25 PRINT "TO BE ROUNDED OFF"
```

```
 30 INPUT N
 35 IF N>INT N THEN GOTO 50
 40 LET R=N
 45 GOTO 100
 50 LET D=N-INT N
 60 IF D>0.5 THEN GOTO 90
 70 LET R=INT N
 80 GOTO 100
 90 LET R=(INT N)+1
100 CLS
110 PRINT N;" ROUNDS OFF TO ";R
120 PRINT
130 GOTO 10
```

59 Two-Digit Round Off

It is possible to round off to the nearest hundredths place. That is, to two digits after the decimal point. Here's how:

Program Listing

```
 10 LET R=Ø
 20 PRINT "GIVE ME A NUMBER TO"
 30 PRINT "MORE THAN TWO DECIMAL PLACES"
 40 PRINT
 50 PRINT "ORIGINAL NUMBER"
 60 INPUT N
 70 LET R=INT(100*N+0.05)/100
 80 CLS
 90 PRINT N;" ROUNDS OFF TO ";R
100 PRINT
110 GOTO 10
```

60 Percent to Decimal

Checking, interest, sales tax, and other financial programs are more "user friendly" if you don't have to make manual conversions in your head. For example, if you know your savings account earns 8 percent interest, and you need to multiply by the decimal value for 8 percent (which is 0.08), it is easier to be able to enter 8 and let the TIMEX figure out the decimal value.

Here's another way to change percentages to decimals inside a program to simplify entry by permitting percents to be entered as simple numbers.

For some examples, try entering a price of 2.50 and a sales tax percentage of 6. Your TIMEX will find the bill totals $2.65. Or try $7.80 and 5 percent tax. The bill will be $8.19. Try $123.75 at 8 percent tax. The bill will total $133.65.

Program Listing

```
 10 LET T=Ø
 20 LET B=Ø
 30 PRINT "PRICE: $";
 40 INPUT P
 50 PRINT P
 60 PRINT"SALES TAX RATE: ";
 70 INPUT R
 80 PRINT R;" PERCENT"
 90 LET T=0.01*R
100 LET B=P+T*R
110 PRINT "BILL TOTALS: $";B
```

61 Every 10th Answer

This program generates a random number in the range of zero to 999. However, it has a difference. It only shows you every tenth number it generates.

Line 20 generates the numbers. Line 40 selects the tenth number from each set.

Program Listing

```
10 LET T=Ø
15 LET V=Ø
20 LET T=INT(1000*RND)
30 LET V=V+1
40 IF (V/10)=INT(V/10) THEN PRINT V,T
50 GOTO 20
```

62 Random Sampler

This program strengthens your confidence in the random number generator built into your TIMEX computer.

It generates 100 numbers between zero and 100 and tells you how many of those are above 49 and how many are below 50. See the sample RUN for several sets of results in our recent test.

Program Listing

```
10 LET N=Ø
15 LET X=Ø
20 LET Y=Ø
25 FOR L=1 TO 100
30 LET X=INT(100*RND)
35 IF X<50 THEN LET Y=Y+1
40 IF X>49 THEN LET N=N+1
45 NEXT L
50 PRINT Y;" YES"
55 PRINT N;" NO"
60 PRINT
65 GOTO 10
```

Sample Run

51	YES	46	YES
49	NO	54	NO
42	YES	52	YES
58	NO	48	NO
56	YES	48	YES
44	NO	52	NO

63 Random Numbers: Zero To Nine

Although you see four program lines below, what we really have here is a very convenient single-line program for you to insert in a larger game or educational-testing program.

Line 20 is the winner here. It prints a random number from zero to nine every time. For your use here, we print that number on the screen. You could just as easily have the computer store that random number in a memory location for later recall and use.

We have added lines 10, 30 and 40 to make your TIMEX show you a whole series of random numbers from zero to nine. Remember, line 20 is the important single-line program element here.

If you would like random numbers in the range from zero to 99, make it 100* in line 20. For zero to 999, use 1000* in line 20.

Program Listing

```
10 RAND
20 PRINT INT (10*RND)
30 PAUSE 100
40 GOTO 20
```

64 Random Numbers: Distribution

Ever wonder how "random" are the numbers generated by the random-number generator in your TIMEX when you use the RND instruction? Try this program.

It generates 100 random numbers in a range from zero to four and counts how many there are of each number between zero and four.

By the way, while it is doing that it will display the message "counting" so you can tell it is working.

At the end of its run, the TIMEX prints a neat chart, on the video display, of results. See our sample run.

Program Listing

```
 10 CLEAR
 20 RAND
 30 LET N=Ø
 40 LET T=Ø
 50 DIM A(5)
 60 FOR L=1 TO 100
 70 PRINT "COUNTING"
 80 LET N=INT(5*RND)
 90 IF N=Ø THEN LET A(1)=A(1)+1
100 IF N=1 THEN LET A(2)=A(2)+1
110 IF N=2 THEN LET A(3)=A(3)+1
120 IF N=3 THEN LET A(4)=A(4)+1
130 IF N=4 THEN LET A(5)=A(5)+1
140 CLS
150 NEXT L
160 FOR L=1 TO 5
170 LET T=T+A(L)
180 PRINT L-1,A(L)
190 NEXT L
200 PRINT "TOTAL",T
```

Sample Run

```
COUNTING

0:  8
1: 10
2: 14
3: 16
4:  6

COUNTING

0:  6
1: 12
2: 11
3: 13
4:  7
```

65 Random Numbers: Averages

This program generates 100 random numbers and totals them. Then it finds the average of all 100 numbers.

In fact, the average number itself is a useful new random number.

Program Listing

```
 10 LET A=Ø
 20 LET N=Ø
 30 LET T=Ø
 40 FOR L=1 TO 100
 50 PRINT "AVERAGING"
 60 LET N=INT(10*RND)
 70 LET T=T+N
 80 CLS
 90 NEXT L
100 LET A=T/100
```

```
110 PRINT "RANDOM NUMBERS TOTAL ";T
120 PRINT "AVERAGE NUMBER IS ";A
```

Sample Run

```
AVERAGING

RANDOM NUMBERS TOTAL 444
AVERAGE NUMBER IS 4.44

AVERAGING

RANDOM NUMBERS TOTAL 424
AVERAGE NUMBER IS 4.24
```

66 Random Numbers: Sorting High/Low

It's important to be able to sort a group of numbers to see what the highest and lowest values are. This program does that.

The random number generator is in line 30. It gives numbers in a range of zero to 999. Line 50 determines the lowest number in the set and line 60 finds the highest number.

Program Listing

```
10 LET H=Ø
15 LET L=Ø
20 LET N=Ø
25 FOR A=1 TO 100
30 LET N=INT(1000*RND)
35 IF A=1 THEN LET L=N
40 IF A=1 THEN LET H=N
45 PRINT "SORTING"
50 IF N<L THEN LET L=N
60 IF N>H THEN LET H=N
70 CLS
```

```
 80 NEXT A
 90 PRINT "LOW NUMBER IS ";L
100 PRINT "HIGH NUMBER IS ";H
```

Sample Run

```
SORTING

LOW NUMBER  IS 32
HIGH NUMBER IS 983

LOW NUMBER IS 14
HIGH NUMBER IS 980

LOW NUMBER IS 17
HIGH NUMBER IS 985

LOW NUMBER IS 9
HIGH NUMBER IS 991

LOW NUMBER IS 1
HIGH NUMBER IS 994
```

Money Matters

67 Money Grows

This section of the book includes a number of programs relating to household money management and to small-business applications. This first program shows you how your money grows when deposited in a savings account at a certain annual interest rate, compounded monthly.

The program will have the computer ask for the initial amount of principal saved by depositing in the account. Then the annual interest rate and the number of months to be displayed. The result of the run is a display of the changing principal as months pass and interest is added on.

Program Listing

```
 10 LET I=Ø
 20 PRINT "PRINCIPAL = $";
 30 INPUT P
 40 PRINT P
 50 PRINT "ANNUAL INTEREST RATE = ";
 60 INPUT R
 70 PRINT R
 80 PRINT "NUMBER OF MONTHS = ";
 90 INPUT M
100 PRINT M
110 FOR Q=1 TO M
120 LET I=(P*(0.01*R))/12
130 LET P=P+I
140 PRINT Q;" MONTH = $";P
150 NEXT Q
160 PRINT
170 IF INKEY$="" THEN GOTO 170
180 CLS
190 GOTO 10
```

68 Shopper's Friend

Take your computer along the next time you go shopping!

This program finds the computer asking for certain information and then telling you which product brand name is the best buy.

The computer wil ask for the brand name of a product, the quantity in the product package, and the price of the package. Then it will ask for the name, quantity and price for a second product.

After digesting all this information, it will tell you the brand name of the best-buy product and show you the unit prices for both brand names so you can agree with the computer's judgment.

For example, suppose you were looking at corn flakes in boxes, one by Post and one by Kellogg. Suppose the Post box contained 24 ounces of flakes and was priced on the grocery shelf at $1.98 while the Kellogg box held 18 ounces and was priced at $1.59. Which would be the better buy based on unit price per ounce of flakes?

Run the data through your TIMEX and you'll find it computes the Post corn flakes to be the best buy with a unit price of 8¢ vs. the Kellogg unit price of 9¢.

After completing the computation, the computer asks for a yes or no answer. Do you have more to be computed?

By the way, if the unit prices turn out to be equal, the computer will say they are equal. If you want no further computations, it will say thanks and quit.

Program Listing

```
 10 PRINT "SHOPPERS FRIEND"
 20 FOR L=1 TO 15
 30 PRINT "*";
 40 NEXT L
 50 PRINT
100 PRINT "FIRST BRAND NAME:"
110 INPUT X$
120 PRINT X$
130 PRINT "QUANTITY: ";
```

```
140 INPUT M
150 PRINT M
160 PRINT "PRICE: $";
170 INPUT N
180 PRINT N
200 PRINT "SECOND BRAND NAME:"
210 INPUT Y$
220 PRINT Y$
230 PRINT "QUANTITY: ";
240 INPUT Q
250 PRINT Q
260 PRINT "PRICE: ";
270 INPUT R
280 PRINT R
300 IF N/M=R/Q THEN GOTO 600
310 IF N/M<R/Q THEN GOTO 400
320 PRINT Y$;" IS BEST BUY"
330 GOTO 500
400 PRINT X$;" IS BEST BUY"
500 PRINT X$;" UNIT = $";N/M
510 PRINT Y$;" UNIT = $";R/Q
520 PRINT
530 STOP
600 PRINT X$;" = ";Y$
610 GOTO 500
```

Sample Run

```
SHOPPERS FRIEND

FIRST BRAND NAME ?
POST
QUANTITY ?
24
PRICE ?
1.98
SECOND BRAND NAME ?
KELLOGS
QUANTITY ?
18
PRICE ?
1.59
```

```
POST IS BEST BUY
POST UNIT=$0.08
KELLOGS UNIT=$0.09

FIRST BRAND NAME ?
BLACKS
QUANTITY ?
100
PRICE ?
25
SECOND BRAND NAME ?
WHITES
QUANTITY ?
50
PRICE ?
12.5

BLACKS = WHITES
BLACKS UNIT=$0.25
WHITES UNIT=$0.25
```

69 Car Payments

Shopping for a new car? Use your TIMEX Computer to compute quickly the potential monthly car payment on various models.

Imagine you want an $8000 car and are prepared to put up $1000 against the purchase. You want to arrange to finance the car for 36 months. You know the current annual interest rate on car loans is 15 percent.

Key in those few numbers and the computer instantly tells you the car payment will be $242.66 per month.

Program Listing

```
10 LET P=Ø
20 PRINT "CAR PAYMENT"
30 FOR L=1 TO 11
40 PRINT "*";
```

```
 50 NEXT L
 60 PRINT
100 PRINT "PURCHASE PRICE ?"
110 INPUT T
120 CLS
130 PRINT "DOWN PAYMENT ?"
140 INPUT R
150 CLS
160 PRINT "NUMBER OF MONTHS ?"
170 INPUT N
180 CLS
190 PRINT "ANNUAL INTEREST ?"
200 INPUT I
210 LET I=(0.01*I)/12
220 CLS
230 LET P=(T-R)*I/(1-1/(1+I)**N)
240 LET P=INT(100*P+.5)/100
250 PRINT "PAYMENT WILL BE"
260 PRINT "$";P;" A MONTH"
```

Sample Run

```
**************
CAR PAYMENT

PURCHASE PRICE $ ?
8000
DOWN PAYMENT $ ?
1000
NUMBER OF MONTHS ?
36
MONTHLY INTEREST ?
15/12

PAYMENT: $ 242.66
```

70 To Nearest 95 Cents

Many companies like to price their goods at a figure ending in 95 cents. For instance, a ten dollar item might be marked $9.95 or $10.95.

Here's a program which demonstrates how to make all prices come out to the nearest 95 cents. See line 40. It merely takes the integer portion of the dollars number and adds 0.95 to it.

Program Listing

```
10 LET P=Ø
20 PRINT "MANUFACTURING COST:"
25 INPUT C
30 PRINT "PRICING MULTIPLIER:"
35 INPUT M
40 LET P=INT (C*M)+0.95
50 CLS
60 PRINT "RETAIL PRICE: $";P
70 PRINT
80 GOTO 10
```

71 To the Nearest Penny

This program is useful when you have a dollar-and-cents figure with more than two decimal places. For example, $151.6972. You need to transform $151.6972 to the more common $151.70

This small program would make a good subroutine in a larger set of instructions. To do so, insert GOSUB at the appropriate place in the larger set of program lines. Modify the line numbers of this small program so the subroutine will be located in an unused position in the larger listing. Change the last line of this small program to RETURN.

Program Listing

```
10 LET R=Ø
20 PRINT "ENTER A NUMBER TO"
30 PRINT "MORE THAN TWO DECIMAL PLACES"
```

```
 40 PRINT "ORIGINAL AMOUNT $ = "
 50 INPUT N
 60 LET R=INT(100*N+.5)/100
 70 CLS
 80 PRINT "TO THE NEAREST PENNY"
 90 PRINT "$";N;" IS $";R
100 PRINT
110 GOTO 10
```

72 Mark Up

Mr. Storekeeper, here's just what you have needed to compute mark ups. This program causes your TIMEX to find the retail price for which your percentage off would give the wholesale cost.

For instance, if you got 40 percent off on an item and paid $60, how much was it priced at, at retail? The answer is $100. To put that another way, if retail price or suggested retail price is $100 and you got 40 percent off at wholesale, what is the wholesale price? The answer is $60.

Try $40 wholesale which is 60 percent off. The answer is $100 retail. Or try $10 wholesale at 90 percent off. Retail would be $100. Or $75 wholesale at 25 percent off gives $100 retail.

Here's a toughie! Try $19.95 wholesale cost. Mark-up percentage is 40. The correct retail answer is $33.25.

Program Listing

```
10 LET D=Ø
20 LET R=Ø
30 PRINT"WHOLESALE COST = $";
40 INPUT W
50 PRINT W
60 PRINT "MARK-UP PERCENTAGE = ";
70 INPUT P
80 PRINT P
90 CLS
```

```
100 LET D=1-0.01*P
110 LET R=W/D
120 PRINT "RETAIL PRICE = $";R
130 PRINT
140 GOTO 10
```

73 Percentage Off

From earlier tips in this book, you know how to make your TIMEX convert percentages to decimals. But what if you want to know "percentage off?"

For example, how much is 40 percent off? This program can be used to interpret 40 percent off and compute the decimal value needed. Try 40 percent off $100. The computer will change 40 percent off into decimal value 0.60. If you multiply 0.60 times $100 you find $60 is 40 percent off $100.

Line 30 makes the important translation.

Program Listing

```
10 PRINT "PERCENTAGE OFF: ";
20 INPUT P
30 PRINT P
40 PRINT "DECIMAL: ";(1-0.01*P)
50 PRINT
60 GOTO 10
```

74 Dollars & Cents

If the result of your computation is a "money" answer, and you don't know whether to display it in dollars or cents, let the computer decide.

This program decides whether to display the output in dollars or cents. Line 90 in the program makes the decision.

Program Listing

```
 10 LET T=Ø
 20 PRINT "QUANTITY = ";
 30 INPUT P
 40 PRINT P
 50 PRINT "TOTAL COST = $";
 60 INPUT C
 70 PRINT C
 80 LET T=C/P
 90 IF T<1 THEN GOTO 130
100 PRINT "EACH COST $";T
110 PRINT
120 GOTO 10
130 LET T=100*T
140 PRINT "EACH COST ";T;" CENTS"
150 PRINT
160 GOTO 10
```

75 Wages & Hours

These useful lines compute total hours worked at regular pay and number of hours worked at time-and-a-half overtime. The computer then finds gross pay and rounds off to the nearest cent.

The program knows that overtime starts after 40 hours. It makes payroll bookkeeping quick and simple.

Program Listing

```
10 LET T=Ø
20 LET W=Ø
30 PRINT "HOURLY PAY RATE = $";
40 INPUT P
50 PRINT P
60 PRINT "NUMBER HOURS WORKED = ";
70 INPUT H
80 PRINT H
90 IF H>40 THEN LET T=H-40
```

```
100 IF H>40 THEN GOTO 140
110 LET W=H*P
120 PRINT "GROSS WAGES = $";W
130 STOP
140 LET W=(40*P)+(T*P*1.5)
150 GOTO 120
```

Sample Run

```
HOURLY PAY RATE = $
5.75
NUMBER HOURS WORKED =
61
GROSS WAGES = $411.125
```

76 Invoicing

There's a lot of repetitious math work to be done before you mail invoices to your customers. This software has the computer collect a few pertinent bits of data from you and then present all the various totals you need to plug into an invoice.

It gives you a total retail price for all goods sold on the invoice, total sales tax if applicable, shipping charges and the grand total amount due you from your customer.

Program Listing

```
 10 PRINT "QUANTITY SOLD: ";
 20 INPUT Q
 30 PRINT Q
 40 PRINT "UNIT PRICE: $";
 50 INPUT P
 60 PRINT P
 70 PRINT "SALES TAX RATE PERCENT: ";
 80 INPUT S
 90 PRINT S
100 PRINT "SHIPPING CHARGES: $";
```

```
110 INPUT H
120 PRINT H
130 LET S=0.01*S
140 LET C=Q*P
150 LET T=C*S
160 LET F=C+T+H
170 LET A=INT(100*C+.5)/100
180 LET B=INT(100*T+.5)/100
190 LET D=INT(100*F+.5)/100
200 CLS
210 PRINT "TOTAL PRICE = $";A
220 PRINT "SALES TAX = $";B
230 PRINT "SHIPPING CHARGES = $";H
240 PRINT
250 PRINT "INVOICE TOTAL = $";D
260 IF INKEY$="" THEN GOTO 260
270 CLS
280 GOTO 10
```

Sample Run

```
QUANTITY SOLD
178
UNIT PRICE = $
55.98
SALES TAX RATE PERCENT =
6
SHIPPING CHARGES = $
100

TOTAL PRICE = $9964.44
SALES TAX = $597.87
SHIPPING CHARGES = $100

INVOICE TOTAL = $10662.31
```

77 Unit Price

Suppose you find 895 green Widgets and buy them for $695. How much did each green Widget cost? Rounded off, $7.77.

Unit price is total price divided by quantity. The quantity can be expressed in weight, total numbers, etc. It works the same whether you are talking about pounds of coffee, yards of concrete, gallons of ice cream, boxes of books, or units of Widgets.

This program asks for the name of the item, quantity purchased and total price paid. It then displays quantity, name, total and unit price.

Program Listing

```
 10 LET U=Ø
 20 PRINT "ITEM NAME IS ";
 30 INPUT N$
 40 PRINT N$
 50 PRINT "QUANTITY OF ITEMS = ";
 60 INPUT Q
 70 PRINT Q
 80 PRINT "TOTAL PRICE PAID FOR ITEMS = $";
 90 INPUT P
100 PRINT P
110 LET U=P/Q
120 PRINT N$;" UNIT PRICE = $";U
```

Sample Run

```
ITEM NAME IS
WIDGETS
QUANTITY OF  ITEMS =
999
TOTAL PRICE PAID FOR ITEMS = $
14653

BEEP WIDGETS UNIT PRICE = $14.667668
```

78 Bubble Sort

Here's a handy routine, accepting lists of up to 10 names and sorting them into alphabetical order. The size of the list is controlled by the DIM instruction in line 20. You can change it.

Note that the screen goes blank while sorting is going on inside your TIMEX so you must be patient. The more names to sort, the longer it takes.

Of course, you can change names to other categories such as items, products, units, etc.

Program Listing

```
 10 CLS
 20 DIM M$(10,10)
 30 FOR L=1 TO 10
 40 PRINT "NAME: ";
 50 INPUT M$(L)
 60 PRINT M$(L)
 70 NEXT L
 80 CLS
 90 LET T=Ø
100 FOR L=1 TO 9
110 IF M$(L)<=M$(L+1) THEN GOTO 160
120 LET E$=M$(L)
130 LET M$(L)=M$(L+1)
140 LET M$(L+1)=E$
150 LET T=1
160 NEXT L
170 IF T=1 TO 10
190 IF M$(L)<>"" THEN PRINT M$(L)
200 NEXT L
```

79 Daily Code

Businesses everywhere are concerned about security. Banks, credit managers, warehousemen, shipping clerks, office managers, retail storekeepers, all need private daily codes for internal use to prevent unauthorized admission to private files, storage areas, financial records.

Now you can use your computer to generate a weekly set of codes for each day. If you feel insecure about a week's list in use, the computer will give you a new set of code numbers in a flash.

The computer generates a table of randomly-selected codes for seven days at a time.

Program Listing

```
 10 GOSUB 300
100 PRINT "SUNDAY:      ";C
110 GOSUB 300
120 PRINT "MONDAY:      ";C
130 GOSUB 300
140 PRINT "TUESDAY:     ";C
150 GOSUB 300
160 PRINT "WEDNESDAY:   ";C
170 GOSUB 300
180 PRINT "THURSDAY:    ";C
190 GOSUB 300
200 PRINT "FRIDAY:      ";C
210 GOSUB 300
220 PRINT "SATURDAY:    ";C
230 STOP
300 LET C=INT(10000*RND)
310 IF C<1000 THEN GOTO 300
320 RETURN
```

Graphics

80 Screen Full

Fill your video screen with square dots, starting from the upper left, moving down and across to the lower right. Line 200 is a freeze frame.

Program Listing

```
 10 CLS
100 FOR X=1 TO 21 STEP 2
110 FOR Y=1 TO 21 STEP 2
120 PRINT AT X,Y;CHR$ 128
130 NEXT Y
140 NEXT X
200 GOTO 200
```

81 Sine Wave

Note that some complex screen art can be created with only a few program lines. You will want to refine these programs to make them even shorter.

Program Listing

```
10 CLS
20 FOR X=1 TO 63
30 LET Y=SIN X + 10
40 PLOT X,Y
50 NEXT X
60 GOTO 60
```

82 Hold That Pose

The illusion of motion is important in computer graphics. But, to make a character move, you have to draw it first!

Here we have created a friendly little stick man. But, having created him on the graphics screen, how can we keep him there? By using a freeze frame as we have done here in line 300.

Line 300 is an endless loop, holding action at line 300 forever, until you press the break key.

Run this program first. Then, go on to Tip Number 83.

Program Listing

```
 10 CLS
 20 FOR X=20 TO 30
 30 PLOT X,35
 40 NEXT X
 50 FOR Y=35 TO 25 STEP -1
 60 PLOT 30,Y
 70 NEXT Y
 80 FOR X=30 TO 20 STEP -1
 90 PLOT X,25
100 NEXT X
110 FOR Y=25 TO 35
120 PLOT 20,Y
130 NEXT Y
140 PLOT 22,33
150 PLOT 28,33
160 PLOT 25,30
170 FOR X=22 TO 28
180 PLOT X,27
190 NEXT X
200 FOR Y=25 TO 5 STEP -1
210 PLOT 25,Y
220 NEXT Y
230 FOR X=20 TO 30
240 PLOT X,5
250 NEXT X
300 GOTO 300
```

83 Okay, Now Wave

This is a development of the stick man drawn and frozen on the screen in Tip Number 82. Here, we add program lines to draw and seem to move the stick man's left arm on the right side of the TV screen. The result is the appearance of arm motion. Very clever, these stick men!

Program Listing

```
 10 CLS
 20 FOR X=20 TO 30
 30 FOR Y=5 TO 35 STEP 10
 40 PLOT X,Y
 50 NEXT Y
 60 NEXT X
 70 FOR Y=25 TO 35
 80 FOR X=20 TO 30 STEP 10
 90 PLOT X,Y
100 NEXT X
110 NEXT Y
120 PLOT 22,33
130 PLOT 28,33
140 PLOT 25,30
150 FOR X=22 TO 28
160 PLOT X,27
170 NEXT X
180 FOR Y=5 TO 25
190 PLOT 25,Y
200 NEXT Y
210 FOR X=30 TO 26 STEP -1
220 UNPLOT X,15
230 NEXT X
240 PAUSE 60
250 FOR X=26 TO 30
260 PLOT X,15
270 NEXT X
280 PAUSE 60
290 GOTO 210
```

84 Create A Table

This program generates a table of values, as a demonstration on how to set up a table on the video display.

Subroutine lines 900 and 910 generate random numbers in the range of zero to 99. Lines 60 and 70 find how many times through the random number generator it takes to get a number greater than 50. The answer is stored in A.

Lines 90and 100 do it again and store the answer in B.

Line 40 prints the table heading and line 110 displays the results.

Program Listing

```
  5 RAND
 10 LET A=Ø
 20 LET B=Ø
 30 LET T=Ø
 40 PRINT CHR$ 166,CHR$ 167
 50 GOSUB 900
 60 IF X>50 THEN LET A=A+1
 70 IF X>50 THEN GOTO 50
 80 GOSUB 900
 90 IF X>50 THEN LET B=B+1
100 IF X>50 THEN GOTO 80
110 PRINT A,B
120 IF T=10 THEN STOP
130 LET A=Ø
140 LET B=Ø
150 LET T=T+1
160 GOTO 20
900 LET X=INT(100*RND)
910 RETURN
```

85 Go To Black

There are ways to black-out your screen without going to the graphics mode. Line 150 is a freeze-frame.

Program Listing

```
 10 CLS
100 FOR L=Ø TO 21
110 FOR C=Ø TO 31
120 PRINT AT L,C;CHR$ 128
130 NEXT C
140 NEXT L
150 GOTO 150
```

86 Centered Message

Think up a message of up to 20 characters and spaces. Type it into your TIMEX in response to the request here at line 110.

The computer will highlight your message by centering it on the video screen.

Use this handy centering program for titles and other important parts of longer programs. It makes a good subroutine.

Program Listing

```
 10 LET A$=""
100 PRINT "MESSAGE:"
110 INPUT M$
200 LET L=LEN M$
210 LET T=L+4
220 FOR Z=1 TO T
230 LET A$=A$+"*"
240 NEXT Z
250 LET P=(40-T)/2
260 CLS
300 PRINT TAB P;A$
310 PRINT TAB P;"* ";M$;" *"
320 PRINT TAB P;A$
```

87 Beautiful Braided Rug

Just goes to show that some of the most attractive computer graphics require some of the shortest programs.

Program Listing

```
10 FOR X=1 TO 20 STEP 2
20 FOR Y=1 TO 20 STEP 2
30 PRINT AT X,Y;CHR$ 8
40 PRINT AT X+1,Y+1;CHR$ 128
50 NEXT Y
60 NEXT X
```

88 Circling Dot

More round graphics on your rectangular picture tube.

Program Listing

```
10 CLS
20 FOR N=1 TO 12
30 LET X=20-(10*COS(N/6*PI))
40 LET Y=10+(10*SIN(N/6*PI))
50 PLOT X,Y
60 NEXT N
70 GOTO 20
```

89 Making Things Move

Movement on the computer display screen is an illusion. As in any television picture, the turning on and turning off of dots in a pattern across a screen can seem to provide motion to an object drawn on the face of the tube.

There are a number of ways to get the look of motion. Let's send a dot across the screen:

Program Listing

```
10 FOR X=Ø TO 63
20 PLOT X,20+SIN X
30 UNPLOT X-1,20+SIN(X-1)
40 NEXT X
50 FOR X=62 TO Ø STEP -1
60 PLOT X,20+SIN X
70 UNPLOT X+1,20+SIN(X+1)
80 NEXT X
90 GOTO 10
```

90 Window Twinklers

Well, what would you call them?

Program Listing

```
 10 CLS
 20 RAND
100 LET X=INT(63*RND)
110 LET Y=INT(43*RND)
120 PLOT X,Y
130 GOTO 100
```

91 Eyeball Scrambler

Blink. Blink. Blink. It's enough to make your eyeballs hurt!

The screen is filled with all the many printable keyboard characters. Some flash on and off. Some do not. Creates a very lively display!

Program Listing

```
10 LET X=INT(30*RND)
20 LET Y=INT(20*RND)
30 LET C=INT(256*RND)
40 PRINT AT Y,X;CHR$ C
50 GOTO 10
```

92 Flashing Dot

Want a blinking dot in the upper left-hand corner of the screen? You've got it!

This program clears the screen and causes that dot to blink. Timing loops in lines 20 and 40 control the length of time the dot is on and off.

See ASCII number list for other characters to display as a dot. We use 128.

Program Listing

```
10 CLS
20 PAUSE 20
30 PRINT AT 0,0;CHR$ 128
40 PAUSE 40
50 GOTO 10
```

93 Snowfall

White flakes sprinkle down the screen, over and over—until you press the BREAK key. It may be useless but it's a lot of fun to watch!

Program Listing

```
100 FOR X=Ø TO 63
110 FOR Y=10 TO 43
120 PLOT X,Y
130 NEXT Y
```

```
140 NEXT X
200 LET X=INT(63*RND)
210 LET Y=INT(43*RND)
220 IF Y<10 THEN GOTO 210
230 UNPLOT X,Y
240 GOTO 200
```

94 Draw A Box I

Now you know how to draw a box!

Program Listing

```
 10 CLS
 20 FOR X=20 TO 30
 30 PLOT X,35
 40 NEXT X
 50 FOR Y=35 TO 25 STEP -1
 60 PLOT 30,Y
 70 NEXT Y
 80 FOR X=30 TO 20 STEP -1
 90 PLOT X,25
100 NEXT X
110 FOR Y=25 TO 35
120 PLOT 20,Y
130 NEXT Y
```

95 Draw A Box II

Lines 100 to 240 draw a box. Line 300 is a freeze-frame device.

Program Listing

```
 10 CLS
100 FOR Y=10 TO 43 STEP 33
110 FOR X=Ø TO 63
120 PLOT X,Y
```

```
130 NEXT X
140 NEXT Y
200 FOR X=Ø TO 63 STEP 63
210 FOR Y=10 TO 43
220 PLOT X,Y
230 NEXT Y
240 NEXT X
300 GOTO 300
```

96 Painting Exercises

Just goes to show that some of the most attractive computer graphics require some of the shortest programs.

Program Listing

```
 10 CLS
 20 RAND
100 LET W=INT(63*RND)
110 LET X=INT(63*RND)
120 LET Y=INT(43*RND)
130 LET Z=INT(43*RND)
140 FOR L=W TO X
150 FOR K=Y TO Z
160 PLOT L,K
170 NEXT K
180 NEXT L
200 GOTO 100
```

97 Draw A Line

There are different ways to draw a line on the Computer screen.

Program Listing

```
10 LET A$=""
20 FOR L=1 TO 30
30 LET A$=A$+"*"
40 NEXT L
50 PRINT A$
```

Program Listing

```
10 FOR L=1 TO 30
20 PRINT CHR$ 128
30 NEXT L
```

Program Listing

```
10 FOR X=1 TO 50
20 PLOT X,10
30 NEXT X
```

Program Listing

```
10 FOR Y=1 TO 25
20 PLOT 20,Y
30 NEXT Y
```

98 Box Message

Let's use low-resolution drawing abilities to create a box on the video display. First, type in this program to create a box. Again, we've added a never-ending loop at line 500 to freeze the picture. Press the BREAK key to end the run.

Program Listing

```
10 FOR X=8 TO 17
20 PLOT X,2
30 NEXT X
```

```
 40 FOR Y=3 TO 9
 50 PLOT 17,Y
 60 NEXT Y
 70 FOR X=3 TO 12 STEP 9
 80 FOR Y=8 TO 13
 90 PLOT X,Y
100 NEXT Y
110 NEXT X
120 PLOT 7,3
130 PLOT 16,3
140 PLOT 6,4
150 PLOT 15,4
160 PLOT 5,5
170 PLOT 14,5
180 PLOT 4,6
190 PLOT 13,6
200 FOR X=3 TO 12
210 PLOT X,7
220 NEXT X
230 PLOT 16,10
240 PLOT 15,11
250 PLOT 14,12
260 PLOT 13,13
270 FOR X=3 TO 12
280 PLOT X,14
290 NEXT X
300 PRINT AT 17,3;"BOX"
500 GOTO 500
```

99 Boxed Title

Here's a neat, different way to place a box around a program title—or anything else you might like to highlight.

Line 310 is a freeze-frame loop, used here so you can review your handywork.

Program Listing

```
10 CLS
```

```
100 FOR Y=15 TO 25 STEP 10
110 FOR X=15 TO 37
120 PLOT X,Y
130 NEXT X
140 NEXT Y
200 FOR X=15 TO 37 STEP 22
210 FOR Y=15 TO 25
220 PLOT X,Y
230 NEXT Y
240 NEXT X
300 PRINT AT 11,11;"TITLE"
310 GOTO 310
```

100 Draw Bar Graphs

Drawing graphs on the video screen are a popular form of communication today. This program establishes a bar graph on the TIMEX display.

We have selected the business-like example, shown here, to demonstrate how you go about setting up a bar graph on the TV screen.

Lines 10 to 80 are used to input data needed to be graphed. We have selected annual profits as the subject for our graph.

Line 10 asks for total profits in the year 1979; line 40 for 1980; etc. For the purposes of our graph, the maximum value you can enter will be 26. When the computer asks for the annual-profits data, give it numbers from 1 to 26.

Line 110 prints the heading.

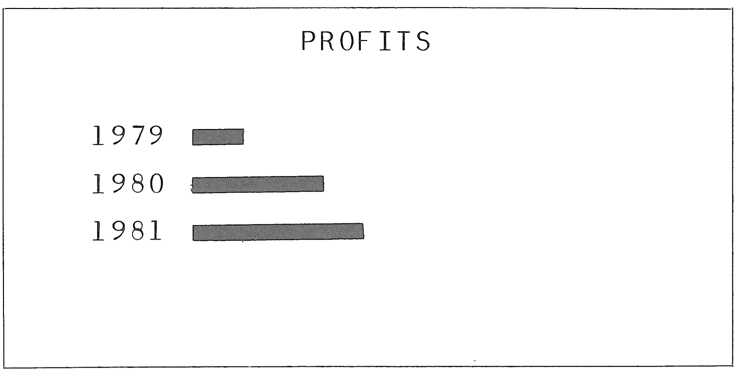
Lines 200 to 240 create the first-year bar on the graph. Lines 300 to 340 create the second-year bar, and so on through line 440 which concludes the last-year bar on the graph.

Program Listing

```
10 PRINT "1979 PROFITS"
20 INPUT A
30 CLS
40 PRINT "1980 PROFITS"
```

```
 50 INPUT B
 60 CLS
 70 PRINT "1981 PROFITS"
 80 INPUT C
 90 CLS
100 LET X$=""
110 PRINT TAB 9;"PROFITS"
200 PRINT AT 3,Ø;"1979 ";
210 FOR L=1 TO A
220 LET X$=X$+CHR$ 128
230 NEXT L
240 PRINT X$
250 LET X$=""
300 PRINT AT 5,Ø;"1980 ";
310 FOR L=1 TO B
320 LET X$=X$+CHR$ 128
330 NEXT L
340 PRINT X$
350 LET X$=""
400 PRINT AT 7,Ø;"1981 ";
410 FOR L=1 TO C
420 LET X$=X$+CHR$ 128
430 NEXT L
440 PRINT X$
```

Sample Run



101 Random Bar Graph

If you don't have any particular data to display, but would like to see how these bar graphs work, try this program.

It uses a random number generator in line 400 to provide data.

Program Listing

```
 10 RAND
 20 LET X$=""
 30 PRINT TAB 9;"PROFITS"
 40 PRINT AT 3,Ø;"1978 ";
 50 GOSUB 400
 60 PRINT AT 5,Ø;"1979 ";
 70 GOSUB 400
 80 PRINT AT 7,Ø;"1980 ";
 90 GOSUB 400
100 STOP
400 FOR L=1 TO INT(27*RND)
410 LET X$=X$+CHR$ 128
420 NEXT L
430 PRINT X$
440 LET X$=""
450 RETURN
```

Appendix

Appendix A: Sinclair BASIC Words

Here is a convenient list, with short explanations, of each of the BASIC language words as used by Sinclair in the ZX-81 and in the TIMEX/Sinclair 1000 computer:

Functions

ABS	absolute value of a number
ACS	arccosine
AND	logical AND
ASN	arcsine
ATN	arctangent
CHR$	changes 0 to 255 number to a character
CODE	number of first character in a string
COS	cosine
EXP	exponent
INKEY$	scans keyboard
INT	integer part of number
LEN	length of a string
LN	natural logarithm
NOT	logical NOT
OR	logical OR
PEEK	look at one memory-address location
PI	3.14159265
RND	random-number generator
SGN	sign of a number
SIN	sine
SQR	square root
STR$	change numerical value to string
TAN	tangent
USR	call machine-language routine
VAL	change string to number
+	addition of number or strings
—	subtraction
*	multiplication
/	division
**	raising to a power
=	equals
>	greater than
<	less than
>=	less than or equal to
<=	greater than or equal to
<>	not equal to

Statements

CLEAR	clears variables
CLS	clears screen
CONT	continue after STOP
COPY	copy TV screen on printer
DIM	dimensions numerical array
DIM $	dimensions string array
FAST	starts fast mode of TV display
FOR	first part of FOR/NEXT loop
GOSUB	jump to subroutine
GOTO	jump to a line
IF	first part of IF/THEN decision maker
INPUT	stops so data can be entered
LET	assigns a value to a variable
LIST	displays program list on TV screen
LLIST	displays program list on printer
LOAD	transfers program from tape into computer
LPRINT	PRINT on printer
NEW	erases all of program memory and variables
NEXT	last part of FOR/NEXT loop
PAUSE	holds TV display on for brief delay
PLOT	blacks in a graphics-screen dot
POKE	writes in one memory-address location
PRINT	displays on TV screen
RAND	reseeds random-number generator
REM	statement ignored during run
RETURN	end of GOSUB, jump back to main routine
RUN	clears variables and starts program action
SAVE	records program from computer onto tape
SCROLL	rolls TV display up a line
SLOW	switches to slower display mode
STEP	optional part of FOR/NEXT loop
STOP	temporary halt in a run
TAB	PRINT at a certain place on TV display
THEN	last part of IF/THEN decision maker
TO	follows FOR in FOR/NEXT loop
UNPLOT	blanks out a graphics-screen dot

Appendix B: Error Messages

This is a list of what Sinclair calls *Report Codes* and what you may refer to as error messages. These are notes to you from the computer about mistakes you may have made and about the status of the computer:

Code Number/ Letter	Description of Error or Status
0	successful completion
1	you have a NEXT without a matching FOR
2	you forgot to use LET or DIM or your FOR variable wrong
3	a subscript is wrong
4	there's not enough memory space for what you want to do
5	the TV screen is full
6	you have computed too large a number
7	you have a RETURN without a matching GOSUB
8	INPUT can only be used in a program line
9	CONT can't restart at STOP
A	you are not using the function correctly
B	an integer is out of range
C	you can't VAL that string
D	you used BREAK to interrupt a run or you are using STOP in an INPUT line
F	there is no such program name

books from ARCsoft Publishers

For the TRS-80 PC-1, PC-2, Sharp PC-1211, PC-1500:

99 Tips & Tricks for the New Pocket Computers (PC-2/PC-1500)		
128 pages	$7.95	ISBN 0-86668-019-5
101 Pocket Computer Programming Tips & Tricks		
128 pages	$7.95	ISBN 0-86668-004-7
Pocket Computer Programming Made Easy		
128 pages	$8.95	ISBN 0-86668-009-8
Murder In The Mansion and Other Computer Adventures		
96 pages	$6.95	ISBN-0-86668-501-4
50 Programs in BASIC for Home, School & Office		
96 pages	$9.95	ISBN 0-86668-502-2
50 MORE Programs in BASIC for Home, School & Office		
96 pages	$9.95	ISBN 0-86668-003-9
Pocket-BASIC Coding Form programming worksheet tablets		
40-sheet pad	$2.95	ISBN 0-86668-801-3

For the TRS-80 Color Computer:

101 Color Computer Programming Tips & Tricks		
128 pages	$7.95	ISBN 0-86668-007-1
55 Color Computer Programs for Home, School & Office		
128 pages	$9.95	ISBN 0-86668-005-5
55 MORE Color Computer Programs for Home, School & Office		
112 pages	$9.95	ISBN 0-86668-008-X
Color Computer Graphics		
128 pages	$9.95	ISBN 0-86668-012-8
The Color Computer Songbook		
96 pages	$7.95	ISBN 0-86668-011-X
My Buttons Are Blue and Other Love Poems		
96 pages	$4.95	ISBN 0-86668-013-6
Color Computer BASIC Coding Form program worksheet tablets		
40-sheet pad	$2.95	ISBN 0-86668-802-1

For the APPLE Computer:

101 APPLE Computer Programming Tips & Tricks		
128 pages	$8.95	ISBN 0-86668-015-2
33 New APPLE Computer Programs for Home, School & Office		
96 pages	$8.95	ISBN 0-86668-016-0
APPLE Computer BASIC Coding Form program worksheet tablets		
40-sheet pad	$2.95	ISBN 0-86668-803-X

For the ATARI 400 and 800 computers:

101 ATARI Computer Programming Tips & Tricks		
128 pages	$8.95	ISBN 0-86668-022-5
31 New ATARI Computer Programs for Home, School & Office		
96 pages	$8.95	ISBN 0-86668-018-7
ATARI Computer BASIC Coding Form program worksheet tablets		
40-sheet pad	$2.95	ISBN 0-86668-806-4

books from ARCsoft Publishers

For the Sinclair ZX-81 and TIMEX/Sinclair 1000 computers:

101 TIMEX 1000 and Sinclair ZX-81 Programming Tips & Tricks		
128 pages	$7.95	ISBN 0-86668-020-9
37 TIMEX 1000 & Sinclair ZX-81 Programs for Home, School & Office		
96 pages	$8.95	ISBN 0-86668-021-7
TIMEX 1000 & Sinclair ZX-81 BASIC Coding Form program worksheets		
40-sheet pad	$2.95	ISBN 0-86668-807-2

For the electronics hobbyist:

25 Quick-N-Easy Electronics Projects		
96 pages	$4.95	ISBN 0-86668-023-3
25 Electronics Projects for Beginners		
96 pages	$4.95	ISBN 0-86668-017-9
25 Easy-To-Build One-Night & Weekend Electronics Projects		
96 pages	$4.95	ISBN 0-86668-010-1

Computer program-writing worksheets:
Each in tablet of 40 sheets with stiff backing

Universal BASIC Coding Form	$2.95
IBM Personal Computer BASIC Coding Form	$2.95
ATARI Computer BASIC Coding Form	$2.95
Pocket Computer BASIC Coding Form	$2.95
APPLE Computer BASIC Coding Form	$2.95
Color Computer BASIC Coding Form	$2.95
TIMEX 1000/Sinclair ZX-81 BASIC Form	$2.95

ISBN: International Standard Book Number

ARCsoft Publishers

Post Office Box 132
Woodsboro, Maryland 21798

101 TIMEX 1000/Sinclair ZX-81 Programming Tips & Tricks

by Edward Page

Here's a giant collection of practical, useful, efficient programming techniques and operating shortcuts right out of a master programmer's notebook. Loaded with hints, secrets, shortcuts, tips, tricks and easy-to-follow instructions, this book tells you how to handle routine programming chores more quickly, do special effects, make your TIMEX/Sinclair 1000 computer and Sinclair ZX-81 computer work for you.

Each of the 101 tips and tricks in this book features a complete, ready-to-type-and-run program which will work all by itself, or as part of a larger set of instructions. All 101 programs in this book have been thoroughly tested on the TIMEX/Sinclair 1000 and Sinclair ZX-81 computers.

Learn insider's how-to secrets for using the exciting BASIC words FOR, NEXT, INKEY$, OR, CHR$, INT, LEN, PEEK, RND, STR$, VAL, CLEAR, CLS, DIM, GOSUB, GOTO, IF, THEN, INPUT, LET, PAUSE, PLOT, PRINT, RAND, RETURN, UNPLOT and many more.

Sections in this book include *Fun & Games, Text On Text, Gee Whiz, Number Crunching, Money Matters, Graphics,* and more. There's even a convenient list of all of the BASIC words and error messages used in the T/S 1000 and ZX-81.

Special techniques are here for graphics, searching, ordering, sorting, error trapping, secret messages, writing and playing games, batting averages, math flasher, ASCII character review, percentages, dollars & cents, decision maker, random numbers, killing time, news ticker, invoicing, wages & hours, passwords, and many, many more.

Programs in this book will run without modification on the TIMEX, Sinclair, MicroAce and other computers using the Sinclair version of the BASIC programming language. The programs also will run, with slight modification to some lines, on TRS-80 models I, II, III, 16, Color, PC-1, PC-2, Apple, Atari 400, 600, 800, Commodore Max Machine, VIC-20, 64, Pet, IBM Personal Computer, Sharp PC-1211, PC-1500, Panasonic, Quasar, Hewlett-Packard, and others.

ARCsoft Publishers
WOODSBORO, MARYLAND

ISBN 0-86668-020-9

101 TIMEX 1000/Sinclair ZX-81 Programming Tips & Tricks Page